UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Matematica e Informatica
Dottorato di Ricerca in Matematica e Informatica XXXII Ciclo

TIZIANA ROTONDO

MULTI-SENSOR DATA FUSION

Tesi di Dottorato di Ricerca

TUTOR:	CHIAR.MO PROF. SEBASTIANO BATTIATO
TUTOR AZIENDALE:	ING. VALERIA TOMASELLI
CO-TUTOR:	CHIAR.MO PROF. GIOVANNI MARIA FARINELLA
COORDINATORE:	CHIAR.MO PROF. GIOVANNI RUSSO

Abstract

Nowadays, thousands of information, such as images, videos, audio signals, sensor data, etc., can be collected by many devices. The idea of multi-sensor data fusion is to combine the data coming from different sensors to provide more accurate information than that a single sensor alone.

Sensors have been proposed to emulate the human capability to combine all senses to capture information. So, the goal of Multimodal Learning is to create models that are able to process information from different modalities, semantically related, creating a shared representation to improve accuracies than could be achieved by the use a single input. In other words, the challenge is constructed an embedding space where objects, which are correlated, are close to them.

Human can also anticipate the future action because our brain is able to decode all information received to understand the future occurrences and to make a decision. The overall design of machines that anticipate future actions is still an open issue in Computer Vision.

To contribute to ongoing research in this area, the goal of this thesis is to analyse the way to build a shared representation related to data coming from different domains, such as images, signal audio, heart rate, acceleration, etc., in order to anticipate daily activities of a user wearing multimodal sensors. To our knowledge, in the state of the art, there are not results on action anticipation from multimodal data, so the prediction accuracy of the tested models is compared with respect to the classic action classification which is considered as a baseline. Results demonstrate that the presented system is effective in predicting activity from an unknown observation and suggest that multimodality improves both classification and prediction in some cases. This confirms that data from different sensors can be exploited to enhance the representation of the surrounding context, similarly to what happens for human beings, that elaborate information coming from their eyes, ears, skin, etc. to have a global and more reliable view of the surrounding world.

Acknowledgements

I would like to express my gratitude to my advisors, Prof. Sebastiano Battiato, Prof. Giovanni Maria Farinella and Ing. Valeria Tomaselli. They have been providing continuous guidance, constructive feedback, and knowledge sharing to me during my whole Ph.D. study. I would like to thank you for allowing me to grow as a research, by encouraging professionalism and independence.

Thanks to the colleagues of the System, Research and Application (SRA) Group of STMicroelectronics. They help me to understand how research should be performed in an industrial context like STMicroelectronics, which sponsored this doctoral fellowship.

Last but not the least, I would like to thank my family. This dissertation would not have been possible without their warm love, continued patience and endless support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, with the advances in technology, we are able to collect many kinds of data. Indeed, the real time information comes from multiple sources such as wearable sensors, smartphones, GPS, etc. Data from different sensors can be exploited to enhance the representation of the surrounding context of a user, similarly to what happens for human beings, that elaborate signals acquired in different ways (eyes, ears, etc.) to have a global and more reliable view of the surrounding world. Humans rarely infer knowledge with only one sense. For instance, humans combine senses as sight, taste and touch to perceive if a food is hot or cold, or in speech recognition, they integrate the information coming from eyes and ears to understand speech. An useful example is given by the McGurk Effect [1], where some people perceive the syllable /da/ by watching the lip movement of the syllable /ga/ and hearing a voice that says /ba/. Since the visual modality combines the information coming from the lip motions and the place of articulation, it can help to discriminate speech with similar sound.

This motivates the study of the intelligent system able to consider multiple signals. Indeed, Artificial Intelligence techniques could be exploited on multiple signals, in order to automatically understand the world around us. The idea of multi-sensor data fusion is to combine the data coming from different sensors to provide more accurate information than that a single sensor alone. Data fusion and multimodal representation ([2, 3, 4, 5, 6, 7, 8]) are challenging tasks in Multimodal Learning, because the different nature of the data is to be considered. The fusion of the data can be done at the following different levels: raw data level, feature level and decision level [9]. In raw data level, raw data from different sensors are combined in order to generate a new modality. The goal of feature level is to fuse the features

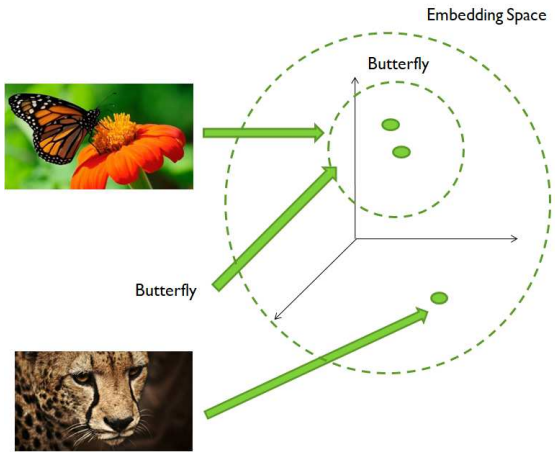


Figure 1.1: Idea of multimodal learning.

extracted from individual sensor modality. Decision level refers to the combination of the classification computed from each modality independently.

To improve these algorithms, a good representation of the multimodal data is also needed. This is not only a theoretical problem but many machine learning approaches are presented in the state of the art. In general, Multimodal Learning aims to build models that are able to process information from different modalities, semantically related, creating a shared representation to improve accuracies than could be achieved by the use a single input. As shown in Figure 1.1, given the images of a butterfly, one of a tiger and the word "butterfly", multimodal algorithm try to project these data in a space, called representation or embedding space, witch takes into account their correlation.

A good model of multimodal learning must satisfy certain properties. In fact the shared representation must be such that resemblance in the embedding space implies that the similarity of the inputs can be easily obtained even in the absence of some modalities. Multimodal learning allows several applications in robotics [10, 11], in surveillance [12, 13] and so on.

In general, each modality is characterized by different statistical properties and hence each one of it can add valuable and complementary information to the shared representation [4]. Moreover, since each input has a different representation and each associated feature vectors is projected in a different subspaces of the embedding space, for a model, it is difficult to find a highly non linear relationship between

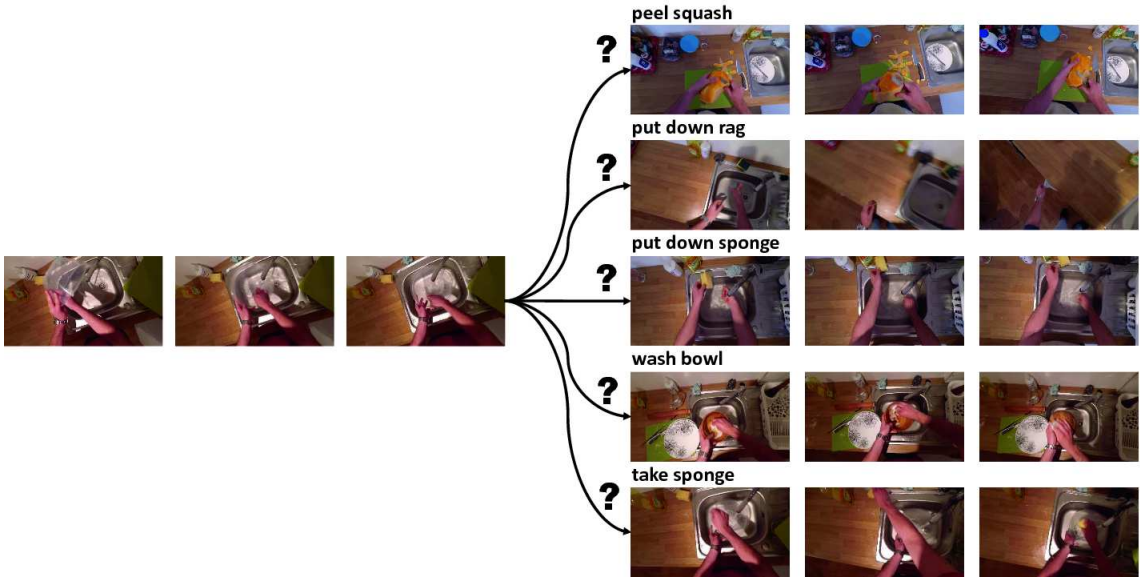


Figure 1.2: Given a video sequence, there could be many possible future activities [18].

different data. So, given the heterogeneous nature of the data, before applying the algorithms of multimodal learning, the features are extracted from each input in different ways. For example, Convolutional Neural Networks (CNNs) [14], such as GoogleNet [15] or VGGNet [16], are used to extract features from images, Mel-frequency cepstral coefficients (MFCCs) [17], spectrogram and in general spectral features are used for audio signals, and so on.

Humans are also able to decode all information received about the environment to understand the future (i.e., what will happen next) in order to make a timely decision. Indeed, starting from past experiences, our brain is able to associate similar semantic events and/or situation in order to generalize and understand the next action to be performed. For example, given a short video or an image, humans can predict what is going to happen in the near future [19], a player can anticipate the trajectory of a ball or a driver can predict the motion of other cars and pedestrian in order to prevent an accident [20], and so on.

In Computer Vision, the design of models to anticipate future actions is still an open challenge. Indeed, as reported in [19], to predict a future action, it is important to identify as much details from the current observations. This is so effective because the human activities are based on sub-activities that are carried out in a sequence. From computational point of view, the anticipation of the next

activity must be done as quickly as possible, in order to create machines, such as robots, that are able to interact with human world reactively. This allows many applications in order to provide adequate assistance to the user. For example, [20] proposed a Recurrent Neural Network model for anticipating accidents in dashcam videos. [21, 22] studied how to enable robots to anticipate human-object interactions from visual input in order to provide adequate assistance to the user. [23, 24, 25] studied how to anticipate human activities for improving the collaboration between human and robot. In [26], the authors propose a new dataset, called Epic-Kitchen Dataset, and action and anticipation challenges have been investigated. However, given the observation of the past, there could be multiple plausible future activities. An example is shown in Figure 1.2 [18]. Given a past sequence, there are many possible future activities, such as peel squash, put down rag, put down sponge, wash bowl and take sponge, which have similar scenario but some of them are less probable to occur than others, such as peel squash and put down rag (see Figure 1.2). So, the generation of the future is really challenging [27].

1.1 Aim and Problem Definition

Images, videos, audio signals, sensor data can be easily collected in huge quantity by different devices and processed in order to emulate the human capability of elaborating a variety of different stimuli. Are multimodal signals useful to understand and anticipate human actions if acquired from the user viewpoint? With this question in mind, this thesis aims to build a shared representation related to data coming from different domains (i.e., images, audio signal, heart rate, acceleration) in order to anticipate daily activities of a user wearing multimodal sensors.

The definition of the problem is really challenging, since it is associated to a creation of a multimodal pipeline which combines the input features in order to anticipate the future action. By employing some deep architectures, the loss function has to take into account the nature of the data in order to have past and future sequences, semantically related, close in the representation space. Moreover, various modalities are usually collected at different sampling frequencies and in many cases with more than one device, therefore, before to extract features, it is necessary to synchronize all the modalities in order to have all of them properly aligned. Most

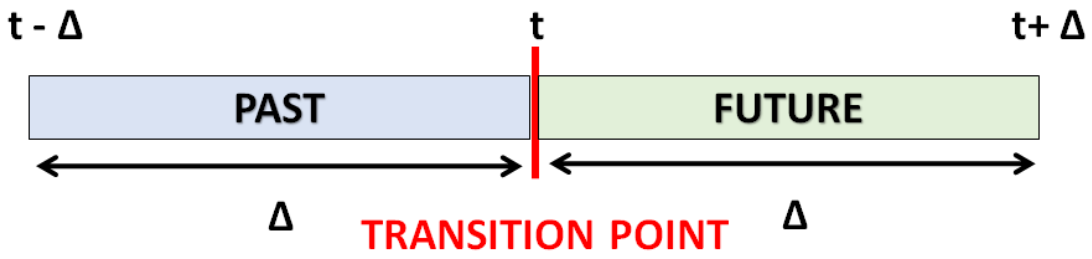


Figure 1.3: Main idea of the "cut" between sequences employed to model the transition among different activities.

of the datasets available in literature contain data acquired with one sensor only (e.g., a camera). Differently than using single modality, multimodal acquisition, especially when collected by means of multiple devices, need to be synchronized in order to have all the related modalities properly aligned over time. Most of the currently available datasets are designed for activity recognition task, rather than for anticipation. Therefore, for our study was very important to build a new multimodal dataset for action anticipation purpose. To this aim, we spent a lot of time to collect a new dataset for action anticipation purpose which comprises video data, acquired with a smartphone camera, sensor data and audio signals, collected with a board, called BlueCoin[1]. With the partnership of STMicroelectronics, an Android app was developed to synchronized the two different devices. To our experience, we also noted that, for long sequences, there are some synchronization issues between the devices.

Since humans can anticipate the future activity by decoding all information from past experiences and knowledge, in this thesis, we try to predict the future action like a human by training a neural network with past sequences that represent our past experiences.

One of the most important point is the adaptation of the data. Given a sequence of signal overtime related to different activities, we consider the "transition point", defined as the point in time where there is a change on the activities (see Figure 1.3). Just before and after the transition point are hence performed two different activities and the goal is, given a set of multimodal signals related to the activity on the left side of a transition point, to infer the activity which will be performed next

[1]https://www.st.com/en/evaluation-tools/steval-bcnkt01v1.html

(i.e., the activity on the right side of the transition point). In doing the task only the signals of the left side of the sequences are given. Let be t the time related to the change from an activity to another, we wish to consider the window $[t-\Delta, t[$ of the sequence (the past) to infer the activity which is present in the portion $]t, t+\Delta]$ of the sequences (the future), where Δ is the amount of time considered to encode past and future from multimodal signals.

We define the problem as follows. Given the features vector $\mathbf{x}_t$ at time t as input, we want to predict the label of the next activity in $[t, t+\Delta[$.

1.2 Contributions

This thesis makes the following contributions:

- A system to anticipate the future activity is proposed. We propose and compare different neural networks, such as Auto-encoder, Siamese Network and Triplet Network, with Multi-Layer Perceptron and 1D CNN architecture, to address the problem of action anticipation from multimodal data. The considered data fusion approach is based on the concatenation of the features which were separately extracted from each modality.

- A dataset for action anticipation task is collected. This research focuses on daily activity anticipation. We define 7 daily activities that are quite common in an office. To this aim, a smartphone camera and a board, called BlueCoin, are used. Smartphone device was placed in the chest pocket of the subjects whereas the Bluecoin board on the right wrist of the subjects. The activities are collected from 19 participants. The following data modalities are acquired: video, audio, tri-axial, tri-axial gyroscope, tri-axial magnetic field, pressure and temperature.

- The effectiveness of combining data collected from different sensors and of improving the anticipation accuracy of the daily activities is demonstrated. We use Support Vector Machines (SVM) and K-Nearest Neighbors (K-NN) as the classification algorithms to evaluate our approach. Since in the state of the art there are not results on action anticipation from multimodal data, a

comparison between the accuracy values in classification and in prediction is proposed as baseline.

1.3 Thesis Structure

This thesis is organized as follows.

Chapter 2 provides a review of all the building blocks which are used in the proposed method. Moreover, it reviews concept of multimodal learning. In particular, the fusion and the representation of the data will be discussed. A briefly introduction about some adopted systems used in multimodal learning, such as Boltzmann Machine (BM) and Deep Boltzmann Machine (DBM) [28], Convolutional Neural Network (CNN) [14] and Recurrent Neural Network [29] is given. The Chapter also introduces the concept of Human Activity Recognition and its application in Artificial Intelligence. Indeed, Action Recognition, Action Anticipation and Early Action Anticipation belong to this research area, so in this chapter they will be also discussed, along with the related work presenting the state of the art. Some multimodal datasets, presented in the state of the art, which were collected mainly for action recognition task are also introduced.

Chapter 3 describes our multimodal dataset, called ST Multimodal Dataset, collected for our experiments purpose, focusing on definition of the activities, collection procedure and alignment of the data.

Chapter 4 presents a description of the proposed approach used in this thesis and the first results on action anticipation from different types of data and investigates the contributions of our dataset by evaluating the accuracy values of different modality combinations. To this aim, the Stanford-ECM Dataset [30] has been considered and it comprises video, acceleration and heart rate data. The second part of the Chapter analyses the effectiveness of the proposed approach by considering the ST Multimodal Dataset, introduced in Chapter 3. In this case, other modalities, such as audio and gyroscope, are taken into account. Experimental results show the improvement of accuracy values using an higher number of transition from past to future.

Chapter 5 gives a summary of the research and the conclusions of this thesis whereas Appendix A and Appendix B summarize other researches that have been carried out during the three PhD years. .

1.4 Published Papers

Part of the work presented in this thesis is based on these co-authored papers:

- Rotondo, T.; Farinella, G. M.; Giacalone, D.; Strano, S. M.; Tomaselli, V. and Battiato, S. (2019). **Anticipating Activity from Multimodal Signals.** Submitted to IEEE Transactions on Emerging Topics in Computing Special Section on Assistive Computing Technologies for Human Well-Being Journal.

- Rotondo, T.; Farinella, G. M.; Tomaselli, V. and Battiato, S. (2019). **Action Anticipation from Multimodal Data.** In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP, ISBN 978-989-758-354-4, pages 154-161. DOI: 10.5220/0007379001540161.

- Rotondo, T. (2018). **Multi-sensor Data Fusion for Wearable Devices.** In Doctoral Consortium - DCETE, ISBN , pages 22-28.

Moreover, with the partnership of STMicroelectronics, we are thinking to an eventual patent regarding attention mechanisms which are able to weight the different modalities and decide which sensors turn on or turn off.

During the three PhD years also other research areas have been addressed co-authoring the following papers:

- Rotondo, T.; Ortis, A. and Battiato, S. (2019). **Generalised Gradient Vector Flow for Content-aware Image Resizing.** In Proceedings of the 20^{th} International Conference on Image Analysis and Processing.

- Rotondo, T.; M. Farinella, G.; Chillemi, A.; Ferlito, F. and Battiato, S. (2018). **A Digital Countryside Notebook for Smart Agriculture and Oranges Classification.** In Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: ICETE, ISBN 978-989-758-319-3, pages 381-385. DOI: 10.5220/0006845305470551.

Chapter 2

Background and Related Works

To understand the background of the research topic, in this chapter, all the fundamental concepts will be discussed. This thesis proposes a system that is able to combine data coming from different wearable sensors, such as smartphones, wrist-worn sensors, in order to anticipate daily human activities. The review, therefore, focused on multimodal learning and its applications and on human activity recognition with more emphasis on action anticipation.

This chapter is organized as follows. Section 2.1 describes the basic unit of the proposed system used to create a shared representation between different modalities and anticipate the near future action. In particular, Auto-encoder, Siamese and Triplet network will be discussed with a some applications presented in the state of the art. Multimodal Learning is introduced in section 2.2. In particular, the fusion and the representation of the data coming from different domains are treated. These steps are very important to understand how integrate and represent heterogeneous data, in order to capture the correlation between them. After a description of these tasks, some applications of Multimodal Learning, such as image captioning, lipreading, sentimental analysis, are presented. The most important Neural Networks, such as Boltzmann Machines [28], Convolutional Neural Networks [14] and Recurrent Neural Network [29] , used to create a shared representation are described in Section 2.3. All the fundamental concepts about Human Activity Recognition are discussed in Section 2.4. This research area comprises the following topics: Activity Recognition, Action Anticipation and Early Action Anticipation. The main difference of these branches of study is given by the temporal moment in which a decision is taken. Indeed in Activity Recognition, the prediction of the activity is done after the action has been observed whereas, in Action Anticipation and Early

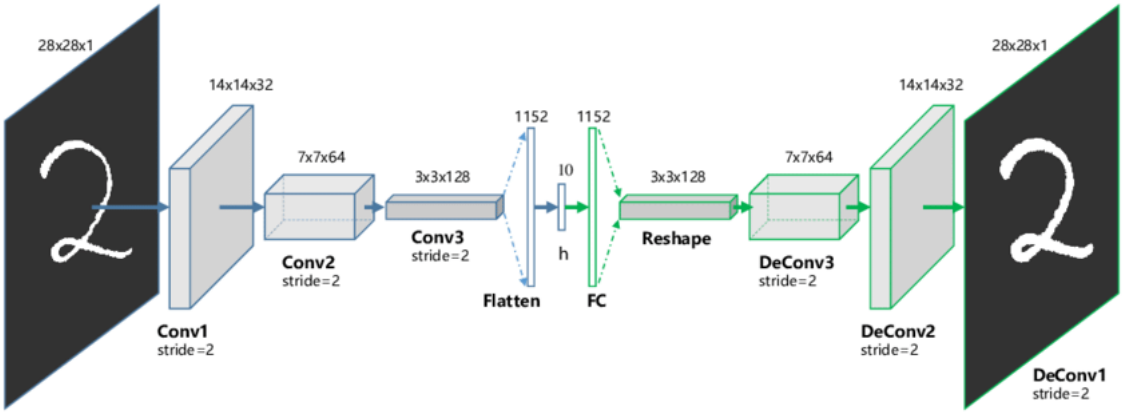


Figure 2.1: Autoencoder Model.

Action Anticipation, the activity is anticipate before the next action starts. Finally, Section 2.5 illustrates some multimodal datasets presented in the state of the art which are created for activity recognition.

2.1 Building Blocks

In this section, the neural networks used in this thesis are described. In particular Auto-encoder, Siamese and Triplet network are considered. Moreover, a briefly introduction of the used classifiers is given.

2.1.1 Autoencoder

Autoencoder has become one of the most popular neural networks for unsupervised learning tasks, such as dimension reduction [31], features extraction [32], image recognition [31, 33], speech recognition [34], and so on. This network [35] applies backpropagation algorithm to learn an approximation to the identity function in order to have output similar to the input, as shown in Figure. In other words, given an input x, it is transformed into $\hat{x}$ thanks to the function f, $\hat{x} = f(x)$, in encoder phase, and then, in decoder phase, $g(\hat{x}) = g(f(x))$ is computed in order to verify the following condition: $g(\hat{x}) \approx x$. To minimize the reconstruction error, it is possible to define the loss function as follows:

$$\mathcal{L} = \mathcal{L}(x, g(f(x))). \tag{2.1}$$

Many variations of this network are presented in literature, such as sparse autoencoder and denoising autoencoder. Sparse auto-encoder [36] is an auto-encoder where the loss function includes a sparsity penalty Ω on the hidden layer. In this way, only a few nodes are activated when a single sample is given to the network. In other word the loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}(x, g(f(x))) + \Omega. \tag{2.2}$$

In [36], the sparsity penalty Ω is defined in various form. It can also be thought as a regularizer term for the copiyng task. In this way, the autoencoder approximate the maximum likelihood training of a model.

Denoising auto-encoder [37] is an autoencoder that accepts as input a corrupted data point and returns as output the original, uncorrupted data point. So the network remove the noise rather than simply reconstruct the input.

In the state of the art, autoencoder was mainly introduced to reduce the dimensions of the input data by extracting the representation of the encoder part. For example in [31], the authors train Restricted Boltzmann Machines and their weights are used to initialize a deep autoencoder to reduce the dimensionality of data. The encoder output is 30-dimensional and the authors also demonstrate that this representation works better than principal components analysis (PCA). Dimension reduction can also improve performance on many tasks, such as semantic hashing which aims to map a document to a binary code [38]. In [34], denoising autoencoder is used to reduce noise in order to clean speech.

In multimodal learning, autoencoder is used in [3] to learn a shared representation between video and audio data. In particular, denoising auto-encoders is used to represent each modality and then fused them into a multimodal representation using another auto-encoder layer. This network is also used, in this paper and in [4], to reconstruct a missing modality. In other words, the model is able to reconstruct the two modalities (e.g. audio and video), given only one as input (e.g. video). The work of [39] introduces a model which uses stacked autoencoders to learn higher-level embeddings from textual and visual input. The two modalities are encoded as vectors of attributes and are obtained automatically from text and images, respectively.

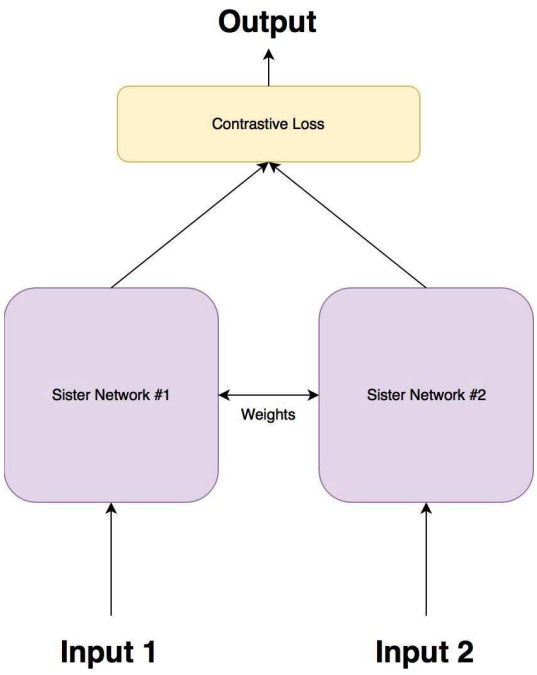


Figure 2.2: Siamese Network Model.

In this thesis, autoencoder is trained like a denoising autoencoder where, given a past clip as input, the model is learned to reconstruct future sequence.

2.1.2 Siamese Network and Triplet Network

Siamese networks [40, 41] consist of twin networks, as shown in Figure 2.2, that perform a non-linear function Φ from the input domain $\mathbf{X}$ to an Euclidean space $\mathbb{R}^n$, i.e. $\Phi : \mathbf{X} \rightarrow \mathbb{R}^n$. These architectures accept two different inputs and share weights, this allows to project, after learning process, two similar images in close points in the feature space because each network computes the same metric function between the highest level feature representation on each stream. So, they are useful to establish and find a relationship between two input data. The outputs of each sub-networks are the feature vectors of the input pair and, at the similarity layer, the distance between the two feature vectors is measured.

To train Siamese Networks, in [42], the contrastive loss function is introduced and it is defined as follows. Let X_1 and X_2 be an input pair and Y a binary label assigned to this pair ($Y = 0$ if X_1 and X_2 belong to the same class, and $Y = 1$

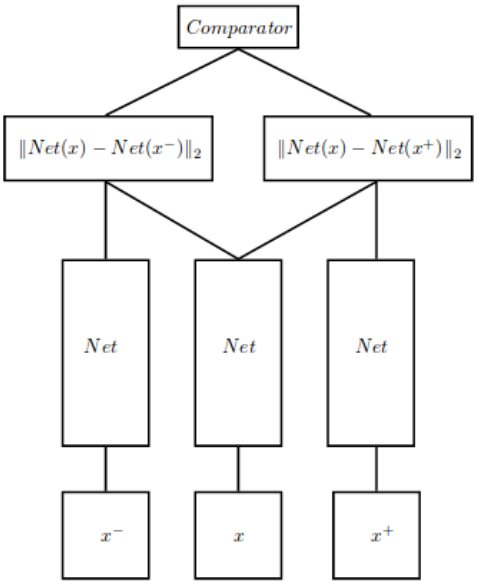


Figure 2.3: Triplet Network Model.

if they have different label), it is possible to define the function that minimize the error as follows:

$$\mathcal{L}(Y, X_1, X_2) = (1 - Y)\frac{1}{2}D(X_1, X_2)^2 + Y\frac{1}{2}\sqrt{\max(m - D(X_1, X_2), 0)} \tag{2.3}$$

where $D(X_1, X_2)$ is a distance between two feature points of X_1 and X_2, $m > 0$ is a margin. In other words, this function is defined as the sum of the loss computed for similar and different pairs. The equation 2.3 allows to have a small distance between similar examples and large distances for different samples.

In [43], an extension of Siamese Network, called Triplet Network, is presented. It comprises three sub-networks, as shown in Figure 2.3, that share the parameters hence accepting three inputs:

- a sample x, called anchor;
- positive clip x^+ which belongs to the same class of x;
- negative sequence x^- that is different from x.

The loss function computes two distances: one considers the representations of x and x^+, whereas the other one is computed from the features extracted with

the network from x and x^-. These distances are computed to assess that the one between x and x^+ is less than one between x and x^-. In this way, the network builds an embedding space where two similar samples are close to each other and at the same time distant from negative samples.

Since Siamese networks and Triplet network are able to recognize if two images, or in general two inputs, are similar or dissimilar, they are used to address many different problems in computer vision and in machine learning. For example, [41, 44, 45] discuss methods which have a very large number of categories but the number of samples for each of them is very small. This is the main characteristics of one shot learning. In other words, the aim of one shot learning is to classify a new example by using a model trained on few samples. In [41], a Siamese convolutional neural network is used in order to learn image representations and reuse network's features without any retraining to recognized characters. The authors of [44] proposed an architecture where each batch point is viewed as a 'query' that ranks the remaining ones based on its predicted relevance to them. The presented model is also able to optimize mean Average Precision over these rankings. The work of [45] defines this task and proposes baseline models where spoken and visual digits are paired.

In [46], a model that determines the location and orientation of a image by matching to a reference database is presented. Moreover, a new loss function which improves the accuracy of Siamese and Triplet Network is also proposed. The work of [47] proposes a social image embedding approach, called Deep Multimodal Attention Networks (DMAN), capture the correlations between image and textual words. The representation of patches in an invariant and discriminative manner is an important task in computer vision. In [48], siamese networks are also used to learn a discriminative representation of small patches captured from different views by training deep convolutional models. Patch representations is also used to track an object in a video [49, 50] where the algorithms find the patch that matches best to the original patch of the target in the frame. More specifically, the methods train a function that compares a pair of images and decide if the two images are similar (the function returns a high score) or different (with a low score). A novel triplet loss is proposed in [51] to extract feature for object tracking by adding it into Siamese network. The matching performance can be improved with the use of triplets in learning local feature descriptors with convolutional neural networks

[52]. These studies can be applied not only for object tracking but also in person re-identification task where the goal is to re-identify the same person across images captured by different cameras with non-overlapping views [53]. The work of [54] proposes a method where features and a similarity metric are learned simultaneously for person re-identification. In [55], a CNN model trained by a triplet loss function which is able to capture both the global full-body and local body-parts features of the input persons is presented. An improvement of the siamese convolutional neural networks is presented in [56] where a matching gate is implemented in order to extract subtle patterns to discriminate hard-negatives from positive pairs.

A similarity metric from data can be also used to address the face recognition task which can be discriminate in face recognition and face verification [57, 58, 59, 60]. In the first case, the aim to recognize a person from a set of face images and find the most similar one to the sample. Given pair of face images or videos, with face verification it is possible to determine if the faces are from the same person or not. Moreover, methods try also to discriminate the images which contain variation of lighting, expression, pose, resolution and background.

In this thesis, Siamese network and Triplet network are used to build an embedding space where past and future sequences semantically correlated are close to each other. These architectures are trained as follows. One stream of the Siamese network processes the past features whereas the other stream processes those related to the future activity, whereas since Triplet Network has three branches, a sub-network processes the past features (x), the second one processes the features related to the possible (true) future activities (x^+) and the last one considers the negative sample related to the false future activities (x^-).

2.1.3 Classifiers

The goal of a classifier is to predict the class/label of given data point. In particular, let be $\mathbf{X}$ the input data and $\mathcal{Y}$ the set of labels, a classifier defines the following function $\Phi : \mathbf{X} \rightarrow \mathcal{Y}$. In this thesis, the proposed approach that anticipates the future action from an observing sequence is evaluate with a SVM classifier for different kernels and with a K-NN classifier for different values of K.

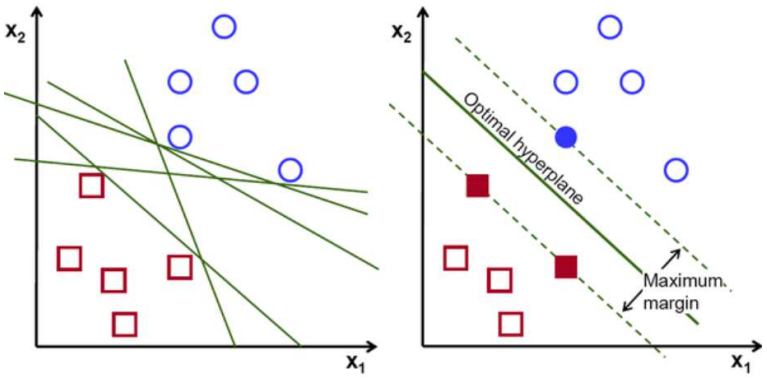


Figure 2.4: Idea of SVM Classifier.

SVM is a parametric model and it tries to solve the following convex optimization problem [61]:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$
$$s.t.\ y_i(\mathbf{w}^T\Phi(x_i) + b) \leq 1 - \xi_i,$$
$$\xi_i \leq 0.$$

where x_i and y_i are training sample and label respectively, $\mathbf{w}$ is the margin, ξ_i is a positive slack variable and C is the penalty parameter of error term which controls the trade-off between $\mathbf{w}$ and ξ_i. Since training samples are projected in a higher dimensional space, the SVM algorithm try to find an hyperplane with the maximal margin which is the distance between the decision boundary and the closest of the data points, as shown in Figure 2.4. An important parameter which plays a main role in the training phase is the kernel. In general, it is defined as $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$. In our experiment, linear kernel $K(x_i, x_j) = \mathbf{x}_i^T\mathbf{x}_j$ and RBF kernel $K(x_i, x_j) = \exp(-\gamma\,||\mathbf{x}_i - \mathbf{x}_j||^2)$, with $\gamma > 0$ kernel parameter, are used.

K-NN [62, 63] is a non-parametric model, indeed its performance depends on the chosen metrics and is not necessary any training. To use this classifier, it is important to choose an appropriate value of K. The reason is explained by Figure 2.5 that shows the idea of K-NN for a set of data with 2 classes. Given a training set and a new sample to classify, the algorithm calculates the distances between the new data and the training samples; the smallest value of distance means that the

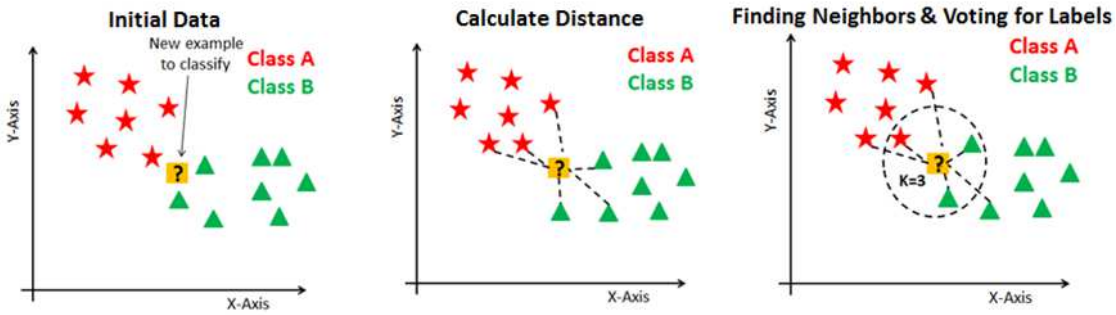


Figure 2.5: Idea of K-NN Classifier.

training sample is closest to the new data, so these samples belong to the same class. In Figure 2.5, K is set to 3 this means that K-NN considers only the closest three samples to new data for classification purpose.

2.2 Multimodal Learning

In the state of the art, there are many papers related to only one modality but we want to extend these concepts to multimodal inputs. This problem is not only theoretical but it has already been dealt with machine learning techniques. The main goal of Multimodal Learning is to build models that process information from different modalities creating a shared representation to improve accuracies that could be obtained by the use a single input. In this section, the problem of the fusion and the representation of the data are discussed.

2.2.1 Fusion of Multimodal Data

Thanks to the many different mobile devices that have been developed, it is possible to simultaneously capture more information of the same situation or scenario. Therefore it is very important to integrate the data coming from different sources in order to have a prediction. Fusion of multimodal data guarantees many advantages. Indeed, since it is possible to have many multimodal data of the same phenomenon, we are able to create a more robust predictions and to capture complementary information. These aspects are similar to the human behaviours. Indeed, humans combine senses to understand and capture all necessary information to make a prediction.

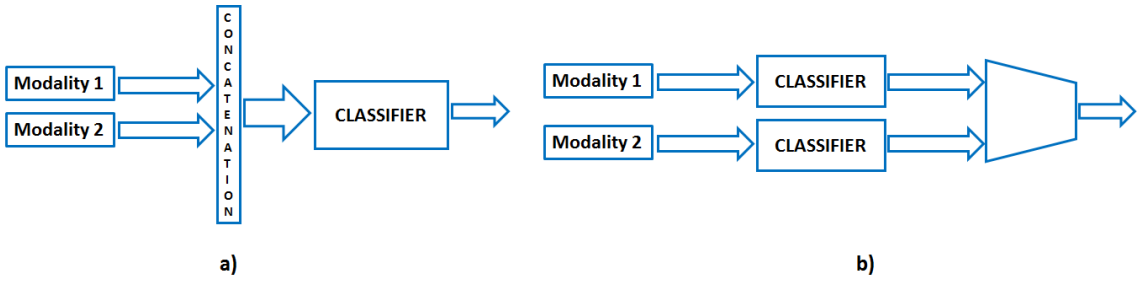


Figure 2.6: a) Early fusion. b) Late fusion.

In Multimodal Learning, many approach of the fusion of the data have been proposed. The main different of these methods is given by the different level of fusion. In [9], the following classification of the fusion approach is given: raw data level, feature level (or early fusion) and decision level (or late fusion). Raw data level refers to the combination of raw data captured from different sources in order to have a new single modality. In [64], a combination of vision depth and inertial sensors is proposed in order to recognize hand gesture.

Feature level or early fusion creates a joint representation of the features that was previously extracted from each inputs, as shown in Figure 2.6 a). So, a single network allows to learns the correlation between each modality, making the training procedure more easier compared to the other strategies. For example, to improve the classification accuracy of a single sensor, in [65], features extracted from the visual modality and accelerometer device is combines by using a Gaussian Mixture Model Bayes classifier.

In decision level or late fusion, shown in Figure 2.6 b), each input is classified and then the results are fused with a fusion mechanism, such as voting scheme [66], a leaned model [67, 68] or averaging [69]. So, for a different modality it is possible to use different classifier but, since the fusion is not compute on raw data, this approach does not learn the low-level relationship between input data. In [70], a late fusion is used to track a human's indoor location. In particular, PIR sensors are used to localize the individual whereas a wearable acceleration sensor is introduced to estimate the human's state.

A combination of early and late technique generates the hybrid fusion strategy [71] which takes advantages from both of them because it combines outputs from early fusion and a classification of each input. An example of hybrid fusion is introduced for speaker identification in [72], where audio signals and visual data are

used as input of a model based on dynamic Bayesian network.

In the state of the art, there are many paper where early fusion and late fusion techniques are compared for visual emotion recognition [73], semantic video analysis [74], etc. In [75], an extension of the aforementioned methods is proposed. It multiplicatively combines the single-source modalities and a set of mixtured source modalities. In order to find accurate fusion architectures for multimodal classification, the work of [76] proposes a generic search space that allows convoluted architectures to take place while also containing the complexity of the problem to reasonable levels. The authors of [77] propose the Low-rank Multimodal Fusion method which performs multimodal fusion using low-rank tensors to improve efficiency. In particular, they demonstrate that this network is able to improve the training and testing efficiency compared to other approaches which performs multimodal fusion with tensor representations.

Recently, attention mechanisms are introduced to weight the contribution of different modalities. In particular, given n arguments $x_1, ..., x_n$ and a context c,the network returns a weighted arithmetic mean of the x_i where the weights are chosen according the relevance of each x_i given the context c [78]. In [79], a Modality Attention mechanism is introduce to computes a set of attention scores which indicate the importance of each modality in order to improve the anticipation of the future action. Attention mechanism is also used to generate caption of a video [80] or a video description [81], where audio features, motion features and video features are used as inputs.

2.2.2 Multimodal Representation

The goal of multimodal representation is to represent data coming from different domains in order to capture the correlation between them. The representation of the data is a challenging task because it must take into account the heterogeneous nature of the data. In general, it is possible to distinguish two types of representation: joint and coordinated representation.

Joint representation aims to project the single modality in an embedding space. The simplest way to represent multimodal data is to concatenate the data features, but other fusion techniques are described in the state of the art. For example, in

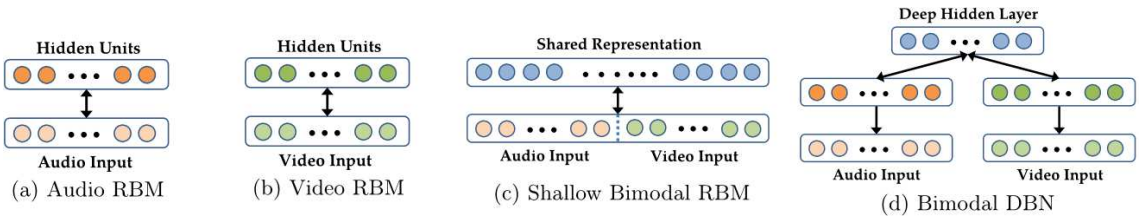


Figure 2.7: RBM Pretraining Models.

[2], to learn the intra-modality and inter-modality dynamics, each modality is represented as tensor and it is constructed using a product from modality embeddings. Neural Network are also used for this task [3, 4, 5]. For example, in [3], video and audio signals are used as input and RBMs are used to build a shared representation, as shown in Figure 2.7. One of the most linear RBM approaches for audio and video is shown in Figure 2.7.a and 2.7.b. The models are trained separately for audio and video data and the resulting probability can be used as a new representation of the data. The concatenation of the inputs is given to the 2.7.c model, but since there are nonlinear correlation between the data, it is difficult for the RBM to provide a multimodal representation. In particular, the units have strong connections between the individual modality and weak connections between units that connect the two inputs. The model in Figure 2.7.c that takes into account the previous ones is considered; in fact, the modalities are trained separately and then the results are concatenated. Another model, called multimodal Deep Boltzmann Machine, is introduced in [4]. The approach is similar to the previous but Deep Boltzmann Machine are used as the basic units for processing data from each modality. The work of [5] proposes a model which is able to construct rich deep representations that are aligned across the following modalities: vision, sound and language. In particular, the authors present a deep convolutional net work that accepts as input either a sound, a sentence, or an image, and produces a representation shared across modalities. Sometimes, it is very important to consider temporal structure of the data during the fusion process. To this aim, in [82] is proposed the Correlational Recurrent Neural Network (CorrRNN) which is a temporal model for fusing multiple input modalities that are inherently temporal in nature. The model is able to capture simultaneously the joint representation and temporal dependencies between modalities, to use of multiple loss terms in the objective function, including a maximum correlation loss term to enhance learning of cross-modal information,

and to use of an attention model to dynamically adjust the contribution of different input modalities to the joint representation. In [83], the Learning Cross-Modality Encoder Representations from Transformers (LXMERT) framework is proposed to learn vision-and-language connections. The model comprises three encoders which are able to capture the relationship between objects, process language and align video and language with a cross-modality alignment. The author of [84] train a multi-task log-bilinear model that compactly encodes word "meanings" represented by each co-occurrence type into a single visual word-vector. In literature, many approaches learn low-level representations, in [85] a joint visual-linguistic model is proposed to learn high-level features without any explicit supervision.

The goal of coordinated representation is to process separately each modality but some similarity constraints are used to coordinate the representations. For example, in [86], to take advantages of intra-modality and inter-modality correlations in order to learn accurate representations, an hashing based orthogonal deep model is learned to build compact binary codes for multimodal representations. Another type of representation is given by the canonical correlation analysis (CCA) [6]. It is a statistical approach that find linear projections of two correlated inputs. An extension of this technique, called kernel canonical correlation analysis (KCCA), is proposed in [8] that finds nonlinear projections of the data by reproducing kernel Hilbert space. This representation is limited by the fixed kernels, so another nonlinear extension of CCA is the deep canonical correlation analysis [7]. This approach learns complex nonlinear transformations of the data in order to have a highly linearly correlated representations.

2.2.3 Applications of Multimodal Learning

In this section, some applications of multimodal learning will be discussed in order to demonstrate the importance of both the combination of the data coming from different domains and the creation of an embedding space where similar objects are close to them.

An important task addressed with multimodal learning is the reconstruction of a missing modality. So, given an information in one modality the task is to map it in a different modality. Foe example, in [3], an autoencoder is used to learn a shared representation between video and audio data and the reconstruction of a

missing modality is proposed. In particular, the model is able to reconstruct the two modalities (audio and video), given only video as input. Moreover, images and texts are used in [4] to the same goal. Another interesting application of Multimodal Learning is introduced in [87] where a system learns to locate image regions which produce sounds and separate the input sounds into a set of components that represents the sound from each pixel. So this system allows to separate sound according to visual data. Moreover, automatic colorization of greyscale images is another task that can be addressed, as shown in [88], where two neural architectures are trained to localize objects mentioned by the captions and properly color them. In general, the reconstruction of a missing modality can be applied in many different field, such as video and image captioning, lipreading and natural language processing, and so on.

Captioning aims to automatically generate a textual description of the actions in a video or in an image. This is a very challenging problem because the models need to understand the visual scene and to identify its salient parts, but also to produce grammatically correct and comprehensive yet concise sentences describing it. One of the first paper that addresses this problem is [89]. The authors combine different architectures to describe an image. Indeed, they use Convolutional Neural Networks to detect image regions, bidirectional Recurrent Neural Networks to produce sentences and a structured objective to align the two modalities through a multimodal embedding. The use of a recurrent model is also proposed in [90] for video captioning. In particular, a variation of a long-short term memory model is introduced to identify discontinuity points between frames or segments and modify the temporal connections of the encoding layer accordingly. Recently, many works of [80, 81, 91, 92] introduce the concept of attention model in order to automatically describe the content of images. In particular, an adaptive encoder-decoder framework is proposed to automatically decides when to look at the image and when to rely on the language model to generate the next word.

A similar task is the reading of the lip movements, called lipreading, where textual modality is decoder from a visual data regarding the movement of a speaker's mouth. It can help human communication and speech understanding in order to prevent, for example, the McGurk effect. To extract features from video, it is important to consider the position, the movement and the shape of the lips in order to

minimize the errors. The work of [93] compares three methods in order to recognize the lip shape. These approaches use hidden Markov models, two of these are top-down approaches therefore derive lipreading features from a principal component analysis of shape or shape and appearance, whereas the last one is a bottom-up method and uses a nonlinear scale-space analysis to form features directly from the pixel intensity. A different technique is presented in [94] where local spatiotemporal descriptors are used to represent and recognize spoken isolated phrases based solely on visual input. A system, called LipNet, is introduced in [95]. This model maps a variable-length sequence of video frames to text with spatiotemporal convolutions, a recurrent network and a connectionist temporal classification loss, trained entirely end-to-end.

In the last decades, with the improvement of social media, it is possible to provide a vary huge quantity of data. This allows to consider problems like the analysis of facial expressions and emotions considering not only images but also combining them with other inputs (e.g. text or audio). In [96, 97], it is demonstrated that the combination of visual, audio and textual features can be effectively used to identify sentiment in Web videos. The work of [98] combines feature of textual, visual, and audio modalities to train a classifier based on multiple kernel learning. In [99], a new sarcasm dataset, called Multimodal Sarcasm Detection Dataset (MUStARD), compiled from popular TV shows is presented. Moreover, it is showed that multimodal models are signicantly more effective when compared to their unimodal variants. Another dataset for sentimental analysis is described in [100] where facial expressions and hand movements during spontaneous spoken communication scenarios are collected from ten actors with different devices placed on face, head and hands.

2.3 Adopted System in Multimodal Learning

In this section, the basic unit systems, presented in the state of the art and used to create a shared representation between different modalities, are described. Many of these are based on the study of different deep networks, starting from Restricted Boltzmann Machines (RBM) [3, 4] to Convolutional Neural Networks (CNN) [30]. Each architecture processes a probability distribution on all multimodal input space.

2.3.1 Boltzmann Machines

The Boltzmann Machines (BM) [28] are networks with a symmetrical connections between binary units, called visible variables $\mathbf{v} \in \{0,1\}^D$ and hidden variables $\mathbf{h} \in \{0,1\}^P$. There are connections between the visible state and the hidden state and between the units of the same type. The energy of the state $\{v, h\}$ is defined as

$$E(v, h; \theta) = -\frac{1}{2} v^T L v - \frac{1}{2} h^T J h - v^T W h, \tag{2.4}$$

where $\theta = \{W, L, J\}$ are the parameters of the model that represent, respectively, the interactions between the visible-hidden, visible-visible and hidden-hidden states. The probability that the model assigns to the visible variable $\mathbf{v}$ is

$$p(\mathbf{v}; \theta) = \frac{p^*(\mathbf{v}, \theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)), \tag{2.5}$$

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)). \tag{2.6}$$

where p^* is the non-normalized probability and $Z(\theta)$ the partition function. Updating the parameters necessary to calculate the log-likelihood with the gradient descent method are obtained from 2.5:

$$\begin{aligned} \triangle W &= \alpha(E_{Pdata}[vh^T] - E_{Pmodel}[vh^T]), \\ \triangle L &= \alpha(E_{Pdata}[vv^T] - E_{Pmodel}[vv^T]), \\ \triangle J &= \alpha(E_{Pdata}[hh^T] - E_{Pmodel}[hh^T]), \end{aligned} \tag{2.7}$$

where α is the learning rate, $E_{Pdata}[\cdot]$ is the data dependency prediction and $E_{Pmodel}[\cdot]$ is the prediction on the model. More details about these terms are given. Starting from 2.5, we know that

$$p(\mathbf{v}; \theta) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)), \tag{2.8}$$

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)) \tag{2.9}$$

$$\exp(-E(\mathbf{v},\mathbf{h};\theta)) = p(\mathbf{v},\mathbf{h};\theta)Z(\theta) \tag{2.10}$$

$$p(\mathbf{h}|\mathbf{v};\theta) = \frac{p(\mathbf{v},\mathbf{h};\theta)}{p(\mathbf{v};\theta)}. \tag{2.11}$$

Log-likelihood function can be easily calculated as follows:

$$\begin{aligned} log(p(\mathbf{v};\theta)) &= log(\frac{p^*(\mathbf{v},\theta)}{Z(\theta)}) \\ &= log(\frac{1}{Z(\theta)}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) \\ &= log(\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) - log(Z(\theta)) \\ &= log(\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) - log(\sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) \end{aligned}$$

Applying partial derivatives both sides and using the equations from 2.8 to 2.11, we obtain:

$$
\begin{aligned}
\frac{\partial}{\partial\theta}(log(p(\mathbf{v};\theta))) &= \frac{\partial}{\partial\theta}(log(\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) - log(\sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))) \\
&= \frac{\partial}{\partial\theta}(log(\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta)))) - \frac{\partial}{\partial\theta}(log(\sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta)))) \\
&= -\frac{1}{\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) \\
&\quad + \frac{1}{\sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta))}\sum_{\mathbf{v}}\sum_{\mathbf{h}}\exp(-E(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) \\
&= -\frac{1}{Z(\theta)p(\mathbf{v};\theta)}Z(\theta)\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) + \qquad (2.12) \\
&\quad \frac{1}{Z(\theta)}Z(\theta)\sum_{\mathbf{v}}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) = \\
&= -\frac{1}{p(\mathbf{v};\theta)}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) \\
&\quad + \sum_{\mathbf{v}}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)).
\end{aligned}
$$

This formulation allows to express the partial derivative of $log(p(\mathbf{v};\theta))$ as:

$$\Rightarrow \frac{\partial}{\partial\theta}(log(p(\mathbf{v};\theta))) = -\sum_{\mathbf{h}}p(\mathbf{h}|\mathbf{v};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) + \sum_{\mathbf{v}}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)).$$

$$\Rightarrow -\frac{\partial}{\partial\theta}(log(p(\mathbf{v};\theta))) = \sum_{\mathbf{h}}p(\mathbf{h}|\mathbf{v};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) - \sum_{\mathbf{v}}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)).$$

Ultimately, the terms $E_{Pdata}[\cdot]$ and $E_{Pmodel}[\cdot]$ have the following expression:

$$E_{Pdata}[\cdot] = \sum_{\mathbf{h}}p(\mathbf{h}|\mathbf{v};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)) \qquad (2.13)$$

$$E_{Pmodel}[\cdot] = \sum_{\mathbf{v}}\sum_{\mathbf{h}}p(\mathbf{v},\mathbf{h};\theta)\frac{\partial}{\partial\theta}(E(\mathbf{v},\mathbf{h};\theta)). \qquad (2.14)$$

The learning algorithm of the BMs requires a very long execution time because

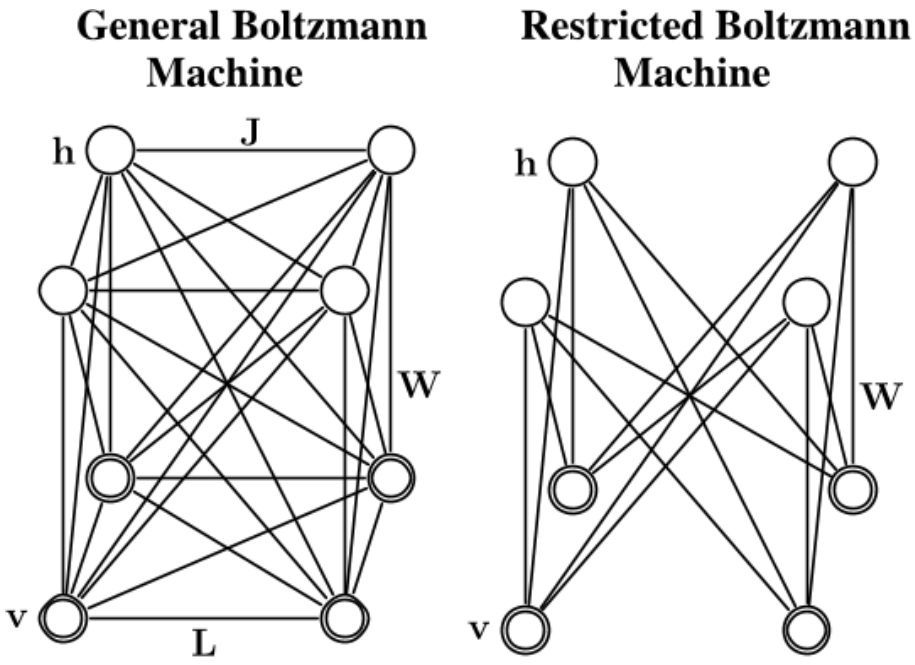


Figure 2.8: Left: A general Boltzmann machine. Right: A restricted Boltzmann machine with no hidden-to-hidden and no visible-to-visible connections.

it is necessary to initialize in a random way the Markov chains to estimate the predictions on the data and on the model. Learning is more effective if you use the Restricted Boltzmann Machines [4, 28] (RBMs). In such models there are connections between the visible layer and the hidden state but there are no connections between variables of the same type, as shown in Figure 2.8.

The parameters L, J are set to zero. In this case the algorithm is efficient using the Contrastive Divergence which provides an approximation of the log-likelihood with a short Markov chain. It is possible to use a stochastic approximation to approximate the prediction of the model. θ_t and X_t, respectively, the parameter and the status are added as follows:

- Given X_t, X_{t+1} is updated by an operator $T_\theta(X_{t+1}, X_t)$ by leaving p_{θ_t} unchanged.
- θ_{t+1} is obtained by replacing the predictability of the intractable model with the prediction against X_{t+1}.

A necessary condition for convergence is that the learning rate decreases as time passes $\sum_{t=0}^{max} \alpha_t = +\infty$ and $\sum_{t=0}^{max} \alpha_t^2 < +\infty$. This is satisfied for $\alpha_t = 1/t$. The described models are the core of the Deep Boltzmann Machines (DBM) [28] which allow us to learn the potential of internal representations and to deal with unlabelled or partially labelled data.

2.3.2 Convolutional Neural Network

Convolutional Neural Networks [14, 29] (CNNs) are networks that process data as multiple arrays by applying convolutional operations. Indeed, audio and sensor can be represented as a one-dimensional signal, audio spectrogram and, in general, image with a 2D signal whereas video as three-dimensional signal. These network are easier to train than the standard ones (such as Multi-Layer Perceptron), because they have few connections between the levels and few parameters. They are characterized by the three following properties:

- each hidden unit is connected to a small local region of the input image and to all the channels of the image;
- many hidden units can share parameters;
- Spatial sub-sampling and pooling operations can be used.

If the input image is gray-scale, the number of channels is one whereas for an RGB image, it is three.

The goal of convolutional layer is to extract the high-level features such as edges, from the input image. Generally, the first layer captures the low-level features such as edges, colour, gradient orientation, etc., whereas with other deep layers, the architecture performs a better fitting to the high-level features. In this way, the network is able to understand images in the dataset. Stochastic Gradient Descent is used to train CNN architectures.

In this thesis, a CNN architecture, called Inception CNN architecture or GoogLeNet [15], is considered to extract features from visual data. It consists of modules that are stacked upon each other, as shown in Figure 2.9. It has 27 layers, 9 of which

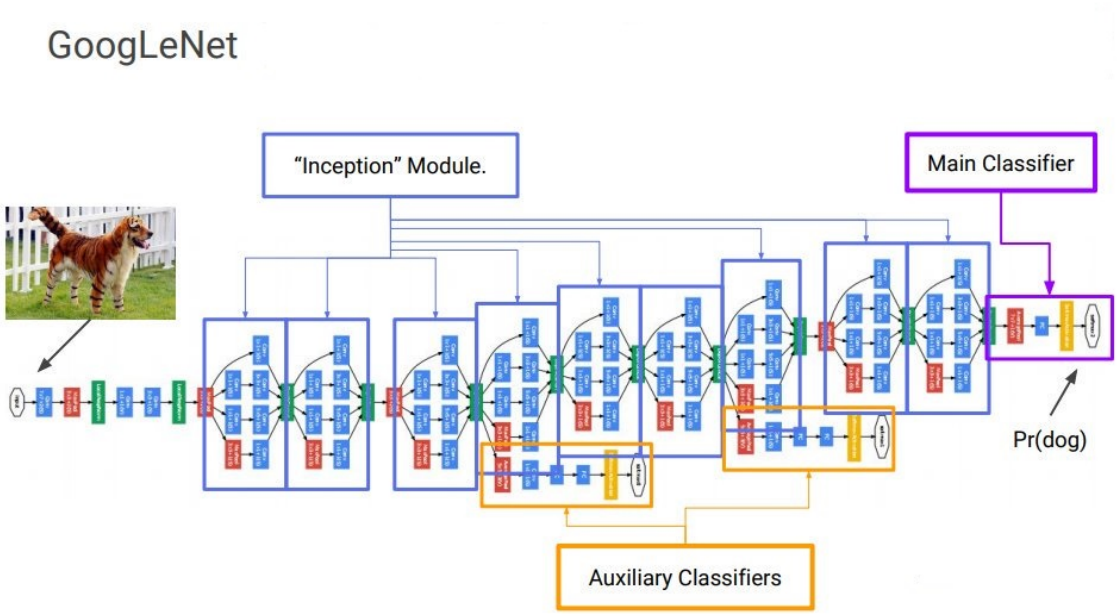


Figure 2.9: GoogleNet Architecture.

are Inception layers. The idea of those layers is to find an optimal local sparse construction. The Inception layer reduces the input size through convolution with the filter size of $1 \times 1, 3 \times 3$ and 5×5. To reduce the dimension of an image and build a sparse representation, before 3×3 and 5×5 convolutions, 1×1 convolutions are used to compute reductions and rectified linear activation is considered.

2.3.3 Recurrent Neural Network

Recurrent Neural Networks (RNNs) [29] are models that process an input sequence one element at a time, keeping in the hidden units a state vector that contains information related to the previous elements of the sequence. The training of these networks is affected by the vanishing and exploding gradient problems that occur when back-propagation error across may time steps [101]. Thus, the calculated and propagated backward gradients tend to increase or decrease at each moment of time, therefore, after a certain number of instants of time, the gradient diverges to infinity or converges to zero. RNNs, as shown in Figure 2.10, is a feed-forward networks where all the layers share the same weights. Since the main purpose of these network

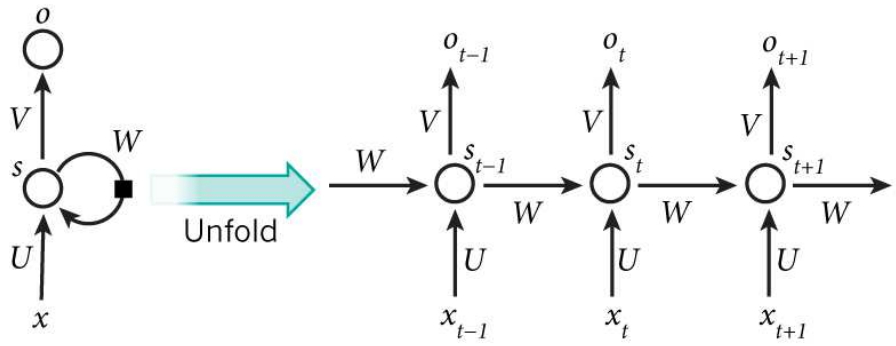


Figure 2.10: RNN Model.

is to learn long-term dependencies, it is difficult to learn to store information for very long. To overcome this problem, long short-term memory (LSTM) networks are introduced where the traditional nodes in the hidden layer are replaced with other units called memory cells. LSTMs are particular RNNs with hidden units that recall previous inputs for a long time. Since, at each instant, these networks take the previous and the current state and the combination of them as input, they decide which information to keep and which to delete from memory.

For example, in [102], a model based on the RNN is proposed with two LSTM layers in order to be able to handle the variations, as follows

$$g_t = Emb(W_{emb}, con(f_{m,t}, f_{o,t})), \tag{2.15}$$

$$h_t = RNN(g_t, h_{t-1}), \tag{2.16}$$

$$p_t = Softmax(W_y, h_t), \tag{2.17}$$

$$y_t = arg \max_{y \in Y} p_t(y), \tag{2.18}$$

where Y is the set of intent indices, p_t is the softmax probability of each intention in Y, W_y is the parameter of the model to train, h_t is the hidden representation coached, g_t is the fixed size of the output of $Emb(\cdot)$, W_{emb} is the parameter of the embedding function $Emb(\cdot)$, $con(\cdot)$ is the concatenation operation and $Emb(\cdot)$ is a linear mapping function.

Policy network π is also introduced to determine when to process an image in a representation of the f_o object. The network continuously observes the movement $f_{m,t}$ and the hidden state of the RNN h_t to be able to calculate $f_{o,t+1}$

2.4 Human Activity Recognition

In the last year, Human activity recognition has become an important task in the Computer Vision community because the recognition of human activities from a series of the human behaviour observations can provide information about the situation and context of a given environment [103]. This allows several applications in different research areas, such as robotics [23, 24, 25], health care [104, 105, 106], surveillance [107, 108, 20], and so on.

The challenging of the task is given by the complexity of human activities. Indeed, human interact with many objects and an activity can be composed by many sub-activities, called atomic activities, which are the basic actions that can not be broken into other ones [109]. For example, cooking eggs may involve taking bowl, opening fridge, taking eggs, closing fridge, etc. Another aspect not less important is represented by the order of the atomic activities, indeed, human can compute them in hundreds different ways (e.g. to cook eggs, we can first open fridge, take eggs, close fridge and then take a bowl).

Real world data are useful to recognize an human activity. With the improvement of the technology, more and more wearable and mobile devices, such smartwatches, smartphone, fitness tracker,etc., have been developed and they allow us to collect millions of data, such as a heart rate, acceleration, temperatures, images, videos, etc., and to track the human behaviour and activities. In vision, it is possible to distinguish the collection of the data in: first person vision and third person vision. With the first one, it is possible to record images and videos by wearing a camera, such as GoPro or Google Glass, so the user's point of view is captured, whereas third person vision is characterized by the fact that images and videos are collected from fixed cameras. So, in first person vision, many details about objects and people are caught with a finer level granularity than third person cameras that capture the high- level information [110].

Another classification is given by the number of modalities which are considered. In the state of the art, there are many paper which consider the problem of human activity recognition from a single input, usually visual modality [21, 107, 111], but it is possible to consider the combination of data, such as skeletons and images [112], audio signals and videos [113, 114], accelerometers [115], and so on.

The study of Human Activity Recognition comprises the following research areas:

Activity Recognition, Action Anticipation and Early Action Anticipation. The aim of the first topic is to recognize a human activity from a sequence that contains the execution of the complete action. Action Anticipation and Early Action Anticipation aim to anticipate the near future activity. The main different between them is that Action Anticipation predict the label after the observation of the past action whereas Early Action Anticipation tries to anticipate the future by observing only an initial portion of data. In the following section each of these research areas are discussed.

2.4.1 Activity Recognition

The goal of activity recognition is to recognize an human action from data that contain the its entire execution. In the last years, many studied have been presented in order to create machines, such as robot, that are able to interact with humans. To this aim, it is very important to implement algorithms that represent and classify the data.

The representation of the data is a challenging task because humans do not always perform an action in the same way and in the same position. In literature, this concept has been discussed. For example, in [116], a movement is defined as motion over time, therefore the authors propose a method that constructs a view-specific representation of movement. In particular, and a binary motion-energy image (MEI), which shows where the motion is computing, a motion-history image (MHI), which defines how the motion is computing, are considered. These two images can be considered as the components of a temporal template which is defines as a vector-valued image that takes into account the position and the motion. To extract space-time features from human action, the work of [117] presents a method which uses the solution to the Poisson equation. The extraction of spatio-temporal information is also considered in [118], where the following descriptors are considered: SIFT average descriptor, trajectory transition descriptor and trajectory proximity descriptor.

Many papers proposed the recognition of the activity from a single input, in particular the visual modality is used. The work of [119] presents a ConvNet architecture for video action recognition. The model combines the appearance and motion pathways of a two-stream architecture by using the multiplicative interactions of space-time features. Many approaches have been also studied in [120] for

extending the connectivity of a CNN in time domain to take advantage of local spatio-temporal information. In [121], a method to recognize daily activities using video from an egocentric camera. The proposed model is able to learn an activity by exploiting the appearance of objects, hands, and actions. The work of [122] presents a deep model for gaze estimation and action recognition in order to determinate what a person is doing and where a human is looking. This is based on the analysis of video collected from a wearable camera. A representation of the first person actions computed from feature trajectories is proposed in [123]. The features are simple to compute using standard point tracking and then are classified with a bag of words classifier.

In multimodal learning, many researches on activity recognition have been presented. A combination of a CNN model and a LSTM is proposed in [112] in order to extract spatial and temporal features from 3D skeletons and images which allow to capture the body motion and the part shape with the purpose of activity recognition. In [30], a model for reasoning on multimodal data to jointly predict activities and energy expenditures is proposed. In particular, for the considered tasks the authors consider egocentric videos augmented with heart rate and acceleration signals. An LSTM is used to take multimodal signals as input composed of the video and the acceleration and returns the activity label and, integrating the heart rate signals, the energy consumption for each frames is also estimated. The work in [102] proposes a on-wrist motion triggered sensing system for anticipating daily intention. They introduces a Recurrent Neural Network (RNN) to anticipate intention and a policy network to reduce computation requirements. In [113], a method based on Hidden Markov Model (HMM) with continuous observation densities is presented in order to recognise and segment actions from continuous audio-visual data. A recognition approach called Non-Markovian Ensemble Voting is proposed by [114]. This method is able to classify multiple human activities in an online fashion. It also can deal with activities that are extended over undefined periods in time. Human action recognition can be addressed by using only sensor data without visual modality. For example, the work of [115] proposes an algorithm that evaluates 20 physical activities from data acquired by five accelerometers worn on different body part. Moreover, other papers combine accelerometer and heart rate data to classify activity [124] and to evaluate energy expenditure [104]. Th work of [125] proposes

a system, called HuMAn, that aims to recognize and classify the human complex activities with a wearable sensing. In [126], a smart assisted living system, that addresses the human intention recognition problem to help elderly people, patients and the disabled, is described. Inertial sensor data are used as input of a Hidden Markov Models (HMM) to classify hand gestures.

Activity recognition can help in medical procedure monitoring to ensuring the quality of a wide variety of healthcare procedures and for the detection of falls of elder people. For example, the work of [127] presents a deep multimodal fusion framework for monitoring and assisting a user perform a multistep medical procedure. The relationships between important life style and health condition is studied, for example to recognize sedentary behaviour [128], to analyze a dairy food [129] or in memory rehabilitation [130]. Accelerometers are also used for fall detection of a elder [131, 132].

2.4.2 Anticipation

Action Anticipation aims to anticipate an activity before it starts by observing the past sequence. The design of machine that predict future occurrences is an open issue in Computer Vision because the generation of the future is not unique. Indeed, given a past sequence, there are many possible future activities which have similar scenario. Therefore a model must be able to identify the details of the current observations in order to anticipate the exact future action.

Since in the state of the art there are not papers on anticipation from multimodal data, in this section, papers that anticipate the next activity from a single modality are discussed.

Many studies in anticipation have been performed to address the data representation task. In [133, 134], the authors propose a study on how to anticipate human actions and objects by learning from unllabeled video. In particular, since representations of future are "easier" to generate than pixels, they proposed a deep network to predict the feature vector representing the images in the future frames. The work of [135] proposes a deep learning framework for human motion capture; it learns a generic representation from a large corpus of previous captured data and generalizes well to new observations. They used an encoding-decoding network that learns to predict future 3D poses from the most recent past. A new feature representation,

named Pooled Time Series, based on pooling feature descriptors over time is introduced in [136]. The work of [111] considers the spatio-temporal prediction problems, where future image-frames depend by control variables, actions as well as previous frames.

The representation is also important to generate future frames from a video or an image. The work of [27] proposes a model for generating the immediate future in unconstrained scenes. The future images are generated by transforming pixels of past images. In [137], the authors use the future poses generated from a Generative Adversarial Network (GAN) to predict the future frames of the video in pixel space. In [138], it is proposed an approach to model future frames from a single input image in a probabilistic manner.

In this thesis, we are interested to consider multimodal inputs to address action anticipation task. So, our goal is to detect and recognize a human action before it happens. The work of [139] proposes a Reinforced Encoder-Decoder (RED) network for action anticipation that takes multiple representations as input and learns to anticipate a sequence of future representations. These anticipated representations are processed by a classification network for action classification. In [19], it is presented a hierarchical model that represents the human movements to infer future actions from a static image or a short video clip. In [140], the authors proposed a method to improve training of temporal deep models to learn activity progression for activity detection and early recognition tasks. Stochastic Context Sensitive Grammar is also proposed in [141] to infer the goal of agents and predict the intended actions. The proposed approach is able to combine events and the temporal relations between the sub-events which are used to discriminate events with similar structures.

Anticipation of the human activities can enable an assistive robot to plan ahead for reactive responses in human environments. In this context, the work in [21] presents an anticipatory temporal conditional random field that models the spatial-temporal relations with object affordance. So, they model the following aspects of activities: the time structure of an activity (which is composed of a sequence of sub-activities), the interdependencies with objects (that allow to perform an activity) and the motion trajectory of the objects and humans. A study on early prediction of the human's motion is also conducted in [24]. Through this study, authors plan robot trajectories that minimize a penetration cost in the human workspace occupancy

while interleaving planning and execution. The importance of the different non-verbal cues for action perception is analised in [25] where the authors propose a robotic system that is able to "read" human action intentions and act in a way that is legible by humans.

Activity anticipation task can also help drivers or autonomous vehicles to prevent a dangerous manoeuvre, such as an accident. This is an extremely challenging task, since manoeuvres are very diverse. Moreover, from human driver experiences, it is necessary to pay attention on scene semantic, object appearance and motion. In [20], a Dynamic-Spatial-Attention Recurrent Neural Network (RNN) for anticipating accidents in dashcam videos is proposed. The authors use an object detector to detect objects, full-frame and object-based appearance and motion features in their model. Another challenges in urban environments is to understand and anticipate pedestrian actions, in particular, at the point of crossing. More approaches use the motion history of road users to predict their future trajectories [142], but, for example, in [143] a RNN architecture with multilevel feature fusion for predicting pedestrian crossing action is proposed.

There are not many large datasets for action anticipation task. The work in [26] proposes a new dataset, called Epic-Kitchen Dataset. The authors show the great potential of huge dataset for first person vision. This dataset is used in [79] where the architecture RU-LSTM, composed by two different LSTMs (one used for encoding and the other for inference), is proposed to address the Epic-Kitchen activity anticipation challenge. A modality attention mechanism is also introduced to weight different modalities in adaptive fashion.

To perform an activity, humans interact with many objects. This problem has been investigated in [22], where the topic of next-active-object prediction from First Person videos is introduced. The authors analysed the role of egocentric object trajectories to anticipate object interactions and propose a suitable evaluation protocol. The authors of [21] address the problem of enabling robots to predict human-object interactions from visual input in order to assist humans in daily tasks.

2.4.3 Early Anticipation

The aim of Early Action Anticipation is to anticipate the future activity by observing only an initial portion of data. This is an useful task because it can help to prevent

many events. Indeed, for example, it is possible to anticipate an accident or a crime before it happens.

To this aim, many papers have been proposed in literature. The work of [144] presents a model, called multiple temporal scale support vector machine (MTSSVM), in order to recognize an unfinished actions. In particular, the approach is able to learn the evolution and dynamics of actions and predict action labels from partially observed videos containing incomplete executions of the actions. Since the anticipation of the future activity is a probabilist method, in [145], the probabilistic way of the anticipation is proposed. Moreover, an activity is defined as an histogram of spatio-temporal features that models how feature distributions change over time. In [108], in order to enrich the feature representations, an approach that reconstructs missing information in the features extracted from partial videos is presented. The work of [146] divides each activity into multiple ordered temporal segments in order to extract spatio-temporal features from each of them. To compute the activity likelihood, sparse coding is applied at each segment, and then the likelihood at each segment is combined in order to obtain a global posterior probability for the activities. In [147], an early detector is proposed to recognize partially events and the method extends the Structured Output SVM. To determinate the relationships between actions and their predictable characteristics, the work of [148] proposes a framework that explores long-duration complex activity prediction. In particular, the authors introduces a method that splits a long activity into a sequence of meaningful action units.

2.5 Multimodal Datasets

In the last years, many action datasets have been released, such as CMU-MMAC Dataset [149], Epic-Kitchen Dataset [26] and Kinetics [150], but they comprise only video data and most of the times they have been designed for action classification, not being tailored for action anticipation task. There are few publicly available multimodal datasets in literature. In the following subsection, some details of publicly available multimodal dataset, created for action recognition task, are given.

2.5.1 Opportunity Dataset

Opportunity dataset [151] contains complex activities with many atomic activities (more than 27,000), collected in a sensor rich environment. Indeed , it was collected by 12 subjects using 15 networked sensor systems, with 72 sensors of 10 modalities, integrated in the environment, in objects, and on the body. Subjects simulated a studio flat with a deckchair, a kitchen, doors giving access to the outside, a coffee machine, a table and a chair. The following sensors are used: 24 custom bluetooth wireless accelerometers and gyroscopes, 2 Sun SPOTs[1] and 2 InertiaCube3[2], the Ubisense localisation system[3] and a custom-made magnetic field sensor. Seven computers acquired the data from specific sensor systems.

2.5.2 Multimodal User-Generated Videos Dataset

Multimodal User-Generated Videos Dataset [152] contains 24 user-generated videos (70 mins) captured using handheld mobile phones both in high brightness and low brightness scenarios (e.g. day and night-time). The video (audio and visual) along with the inertial sensor (accelerometer, gyroscope, magnetometer) data is provided for each video. These recordings are captured using single camera at distinct timings and locations, changing lights and varying camera motions. Each captured video was manually annotated to get labels for camera motions (pan,tilt, shake) at each second. The ground-truth labels are included in the dataset.

2.5.3 CMU-MMAC Dataset

The Carnegie Mellon University Multi-Modal Activity Database (CMU-MMAC) [149] contains multimodal measures of the human activity of subjects performing the tasks involved in cooking and food preparation. 55 subjects have been recorded, cooking the following five different recipes: brownies, pizza, sandwich, salad and scrambled eggs. The dataset comprises the following modalities:

- Video:

[1] https://www.sunspotdev.org/
[2] https://www.vrealities.com/products/head-trackers/inertiacube3-ic3
[3] https://www.ubisense.net/

  - Three high spatial resolution (1024 × 768) color video cameras at low temporal resolution (30 Hertz).
  - Two low spatial resolution (640 × 480) color video cameras at high temporal resolution (60 Hertz).
  - One wearable low spatial resolution (640×480) camera at low temporal resolution (12 Hertz).

- Audio:
  - Five balanced microphones.
  - Wearable watch.

- Motion capture: A Vicon motion capture system[4] with 12 infrared MX-40 cameras. Each camera records images of 4 megapixel resolution at 120 Hertz.

- Five 3-axis accelerometers and gyroscopes.

2.5.4 Stanford-ECM Dataset

Stanford-ECM Dataset [30] comprises 31 hours of egocentric video (113 videos) synchronized with acceleration and heart rate data. The video and triaxial accelerations were captured with a mobile phone equipped with a 720 × 1280 resolution camera at 30fps and 30Hz, respectively. The lengths of the videos range from 3 minutes to about 51 minutes. The heart rate was collected with a wrist sensor every 5 seconds (0.2 Hz). These multimodal data were time-synchronized through Bluetooth. Cubic polynomial interpolation was used to fill any gap in heart rate data. Finally, data have been aligned considering millisecond level at 30 Hz.

The activity classes of Stanford ECM-Dataset are listed in Table 2.1. There are 24 classes in total. "Background" is a miscellaneous activity class which includes activities such as taking pictures or parking a bicycle. The dataset has also an additional class, *unknown*, that is related to part of the data before or after an action occurs.

[4]https://www.vicon.com/

Activity	*Activity*
1.BicyclingUphill	13.Shopping
2.Running	14.Strolling
3.Bicycling	15.FoodPreparation
4.PlayingWithChildren	16.TalkingStanding
5.ResistanceTraning	17.TalkingSitting
6.AscendingStairs	18.SittingTasks
7.Calisthenics	19.Meeting
8.Walking	20.Eating
9.DescendingStairs	21.StandingInLine
10.Cooking	22.Riding
11.Presenting	23.Reading
12.Driving	24.Background

Table 2.1: Activity classes of Stanford-ECM Dataset.

2.5.5 University of Texas at Dallas Multimodal Human Action Dataset (UTD-MHAD)

The UTD-MHAD dataset [153] was collected with a Microsoft Kinect sensor and a wearable inertial sensor in an indoor environment. The collected activities are 27 and are listed in Figure 2.11. These actions were performed by 8 subjects and each of them repeated each action 4 times. The dataset comprises 861 data sequences and the following modalities: RGB videos, depth videos, skeleton joint positions, tri-axial acceleration and tri-axial rotation signals. The wearable sensor was placed on the right wrist of the subject from action 1 to 21 to collect the hand motions and on right leg from action 22 to 27 to detect the leg movement.

2.5.6 Multimodal Egocentric Activity Dataset

Multimodal Egocentric Activity dataset [154] contains 20 distinct life-logging activities performed by different human subjects which can be grouped in the following 4 sets: 'Ambulation', 'Daily Activities', 'Office Work' and 'Exercise', as shown in Figure 2.12. Each activity category has 10 sequences of a duration of 15 seconds. This dataset comprises the following modalities: video, accelerometer, gravity, gyroscope, linear acceleration, magnetic field and rotation vector. Google Glass allows to record egocentric video and sensor data simultaneously in a synchronized fashion.

	Wearable inertial sensor on right wrist	
1	*right arm swipe to the left*	*(swipe_left)*
2	*right arm swipe to the right*	*(swipe_right)*
3	*right hand wave*	*(wave)*
4	*two hand front clap*	*(clap)*
5	*right arm throw*	*(throw)*
6	*cross arms in the chest*	*(arm_cross)*
7	*basketball shoot*	*(basketball_shoot)*
8	*right hand draw x*	*(draw_x)*
9	*right hand draw circle (clockwise)*	*(draw_circle_CW)*
10	*right hand draw circle (counter clockwise)*	*(draw_circle_CCW)*
11	*draw triangle*	*(draw_triangle)*
12	*bowling (right hand)*	*(bowling)*
13	*front boxing*	*(boxing)*
14	*baseball swing from right*	*(baseball_swing)*
15	*tennis right hand forehand swing*	*(tennis_swing)*
16	*arm curl (two arms)*	*(arm_curl)*
17	*tennis serve*	*(tennis_serve)*
18	*two hand push*	*(push)*
19	*right hand knock on door*	*(knock)*
20	*right hand catch an object*	*(catch)*
21	*right hand pick up and throw*	*(pickup_throw)*
	Wearable inertial sensor on right thigh	
22	*jogging in place*	*(jog)*
23	*walking in place*	*(walk)*
24	*sit to stand*	*(sit2stand)*
25	*stand to sit*	*(stand2sit)*
26	*forward lunge (left foot forward)*	*(lunge)*
27	*squat (two arms stretch out)*	*(squat)*

Figure 2.11: Activities of UTD-MHAD dataset.

The video was collected with a 1280×720 resolution and 29.9 fps whereas the sensor data frequency is 10Hz.

2.5.7 Daily Intention Dataset- Object Interaction Dataset- Hand Motion Dataset

In [102], the authors collected three datasets: Daily Intention Dataset, Object Interaction Dataset and Hand Motion Dataset.

The first was used for training model to predict the future and they selected 34 daily intentions. Each of this is associated with a motion and an object. The video was collected with a 640×480 resolution.

The second dataset was used for pre-training an encoder to recognize daily object categories. 50 object categories are selected and they collected 940 videos, recording the way an object instance is interacted by a user's hand.

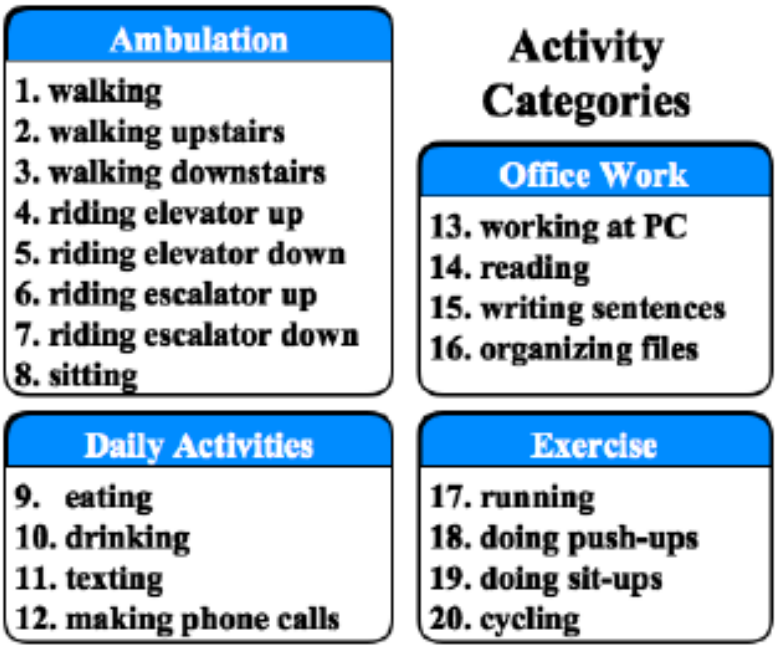


Figure 2.12: Activities of Multimodal Egocentric Activity dataset.

The last one was used for pre-training an encoder for recognizing motion. In this case, eight users collected 609 motion sequences from the right hand and one user collected 36 motion sequences from left hand.

2.5.8 50 Salads Dataset

50 Salads dataset [155] is a multimodal dataset of activities that involve manipulative gestures. It captures 25 people preparing 2 mixed salads and comprises 4 hours of annotated accelerometer data captured at 50Hz, RGB and depth video data collected at 30 Hz and with a 640×480 resolution camera. Accelerometers are attached to a knife, a mixing spoon, a small spoon, a peeler, a glass, an oil bottle, and a pepper dispenser. Two type of annotation are included, indeed, an activity corresponds to either of two levels of granularity, high-level activity and low-level activity. In total, 966 activity sequences are annotated.

2.6 Summary

In this chapter, all the fundamentals of the building blocks of the proposed approach have been discussed. In particular, Auto-encoder, Siamese Network and Triplet Network are introduces and a summary of used classifiers is given. Moreover, some challenging tasks, such as the fusion and the representation of data coming from different domains, and some application of multimodal learning have been introduces.

A state of the art of Human Activity Recognition is also proposed. The review has shown that this research area involves three main challenges in Computer Vision: Activity Recognition, Action Anticipation and Early Action Anticipation. The main difference of these branches of study is given by the temporal moment in which a decision is taken.

Chapter 3

ST Multimodal Dataset

In machine learning, data represent one of the most important aspects to train algorithms. Most of the datasets available in literature contain data acquired with one sensor only (e.g., a camera). Differently than using single modality, multimodal acquisition, especially when collected by means of multiple devices, need to be synchronized in order to have all the related modalities properly aligned over time. Most of the currently available datasets are designed for activity recognition task, rather than for anticipation. Therefore, for our study was very important to build a new multimodal dataset for action anticipation purpose.

In this chapter, we describe our dataset collected for action anticipation task, called ST Multimodal Dataset. The datasets, presented in Section 2.5, were created for activity recognition purpose and some of them can be used for other tasks, such as automatic segmentation, sensor selection and so on. The goal of this thesis is to anticipate the future activity before it starts from multimodal data. So, it is very important to consider a specific multimodal dataset for action anticipation purpose. Moreover, since the problem is defined as the detection of the transition point in order to discriminate past and future sequences, we need more transitions from an activity to another in order to train a neural network for action anticipation task.

To this aim, we collected sequences of activities to create the ST Multimodal Dataset, which is, to our knowledge, one of the first datasets for activity anticipation from multimodal sources, even if it can be used for other different purposes, such as semantic annotation, sensor recognition, object recognition, action recognition, action anticipation from a single modality, and so on. We grabbed video sequences from a smartphone camera as well as different signals from a board, called BlueCoin, built by STMicroelectronics. In particular, data modalities captured by the

BlueCoin board include: audio, tri-axial acceleration, tri-axial gyroscope, tri-axial magnetic field, pressure and temperature. The videos were collected with a resolution of 720×1280 at 29.94 fps, inertial sensors frequency was 52.63 Hz whereas the sampling rate for the audio signal was 32 KHz. Inertial sensor of the BlueCoin recorded data every 19ms. The lengths of the recorded sequences range from 29" to 2' and 59". These characteristics are summarized in the Figure 3.1.

Video	
Resolution:	720×1280.
Frame Rate:	29.94 fps.
Length:	29 sec to 2:59.

Audio	
Number of channels:	4.
Sampling Rate:	32 KHz.
Length:	29 sec to 2:59.

Sensor Data	
Sensor Data:	Accelerometer (X,Y,Z), Gyroscope (X,Y,Z), Magnetic Field (X,Y,Z), Pressure, Temperature.
Sampling Rate:	52.63 Hz.

Figure 3.1: Summary of main characteristics of each modality.

The length of recordings is always below 3:00 minutes, because we noted that, for longer sequences, some synchronization issues between the smartphone and the BlueCoin arised.

We carefully analyzed all the different alternatives for placing the smartphone and the BlueCoin on the user's body, paying attention to the activities of interest by the ST's applicative context (e.g., office). Figure 3.2 shows the sensor placement we chose. The mobile camera was placed in the chest pocket of the subjects, to collect egocentric video, whereas, since most of the considered daily activities were performed with right hand (such as scroll down the page with a mouse), the BlueCoin board was placed on the right wrist of the subjects. The smartphone and the BlueCoin were time-synchronized through Bluetooth.

3.1 Activity definition

The multimodal dataset has been designed for action anticipation focusing on office environment considering the following activities that are quite common in such context:

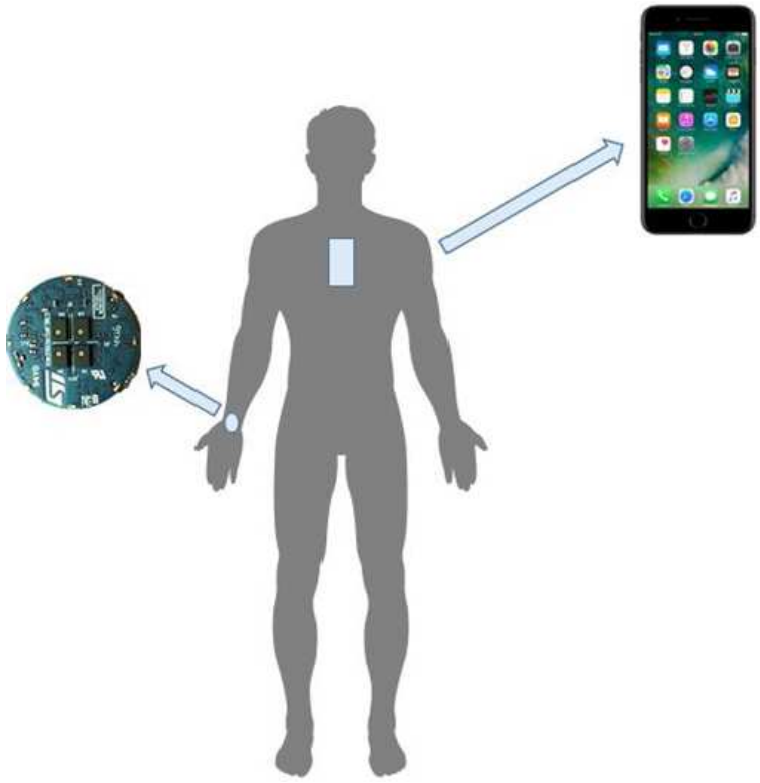


Figure 3.2: Sensors position. Smartphone camera was placed in a chest pocket whereas Bluecoin board on the right wrist of subjects.

Desk is a miscellaneous activity class, such as open/close file, open/close book, drink water, eat, charge the phone and all activities that can be done at a desk by a user with exception of reading and typing that are considered in separate classes;

Reading comprises reading a book, a paper, a journal, a text file, a web page or an email when the user is seated in a desk;

Sitting is related to the movement from standing to sitting (i.e., up-to-down);

Stairs includes sequences of the user going up or down on stairs;

Standing is related to the movement from sitting to standing (i.e., down-to-up);

Typing includes sequences of the writing using a keyboard;

Walking is related to sequences of user walking excluding going up/down from stairs.

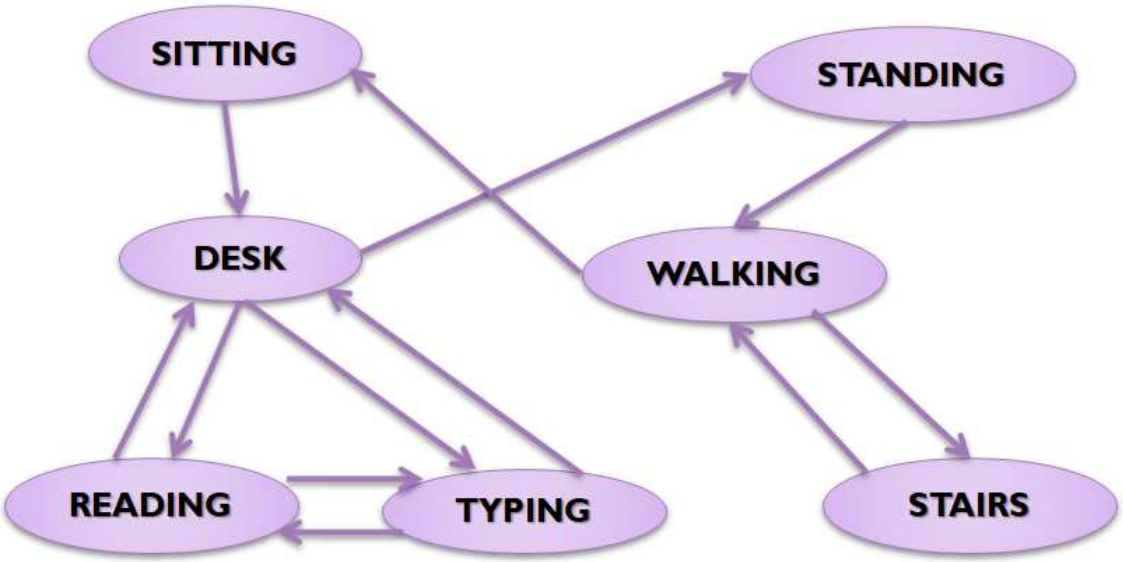


Figure 3.3: Scheme of activities interaction.

The considered activities could be organized in a graph where it is possible visualize the activities interaction. This is shown in Figure 3.3. Some transitions are trivial such as from Standing to Walking or from Sitting to Desk, but some other are challenging; for example, from Desk there are three possible futures, i.e. Typing, Reading or Standing. The following twelve past/future transitions are considered:

- Sitting / Desk
- Desk / Reading
- Reading / Desk
- Reading / Typing
- Typing / Reading
- Desk / Typing
- Typing / Desk
- Desk / Standing
- Standing / Walking

- Walking / Sitting
- Stairs / Walking
- Walking / Stairs

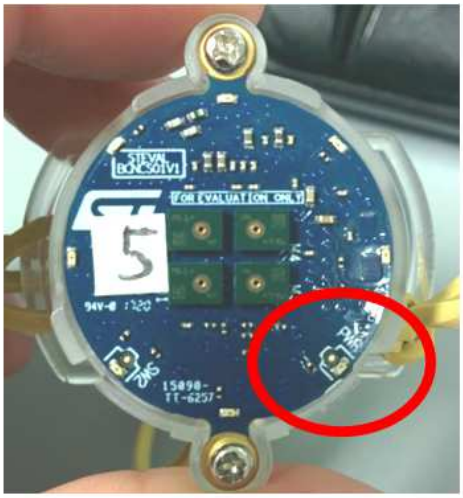
To introduce variability in the dataset, the data have been collected in two different places: STMicroelectronics offices and University of Catania offices. The multimodal sequences were collected with the help of 19 different subjects (5 females and 14 males). Each subject performed each transition defined above about 20 times. This allowed to collect a dataset with large variations, because every subject performed the same activity at different speeds and each activity was repeated many times.
Before data acquisition, the collection procedure was explain to each participant (some details will be given in section 3.2) and we encouraged them to perform each activity in a natural way.

3.2 Collection Procedure

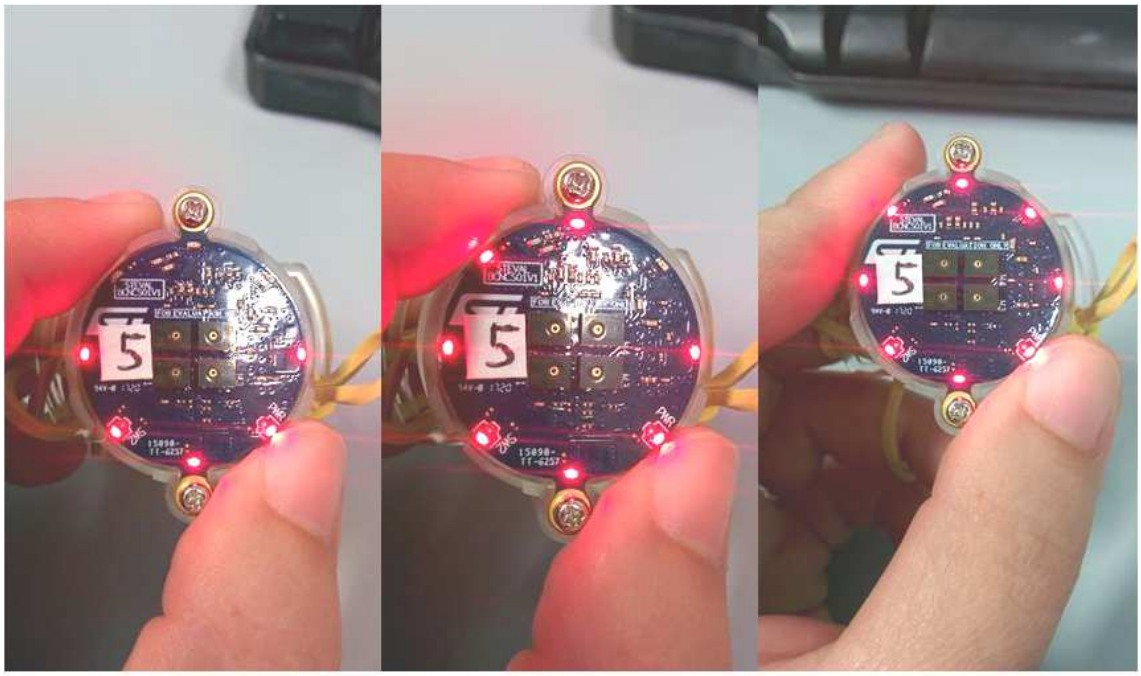
At the beginning of first recording session, the BlueCoin sensor was turned on by keeping the power (PWR) button pressed until all the red LEDs were on, as shown in Figure 3.4.

To collect videos and labels, an Android app, called Multimodal Log, was developed, as shown in Figure 3.5. By clicking on "Start Scanning", the scan for the identification of Bluecoin board begins and in "'Device List" you can select the board to use (E.g. in Figure 3.5 a device, which has been called "Mauro", is detected). After selection, the user can insert his data, such as name, age, height, weight and gender, whereas touching on "Activity", the user can choose the activities to be acquired. By clicking on "START", a recording session starts and the connection with the BlueCoin device is established through Bluetooth, whereas touching on "STOP" the session ends and the connection with the device is interrupted.

At the end of each acquisition, the Bluecoin sensor saves a *.aei* file which contains audio and sensor data and a MATLAB code is used to extract the audio and sensor data. In particular, *.wav* file and *.csv* file are saved; the first one contains the audio clip whereas the second one the raw sensor data. Figure 3.8 presents an example



a)



b)

Figure 3.4: Bluecoin setup.

of Bluecoin sensor data. Each column represents a different type of data of each modality and the measurement unit in parentheses. So, the first column represents the timestamp, acceleration along three axis from 2^{nd} to 4^{th} column, gyroscope along

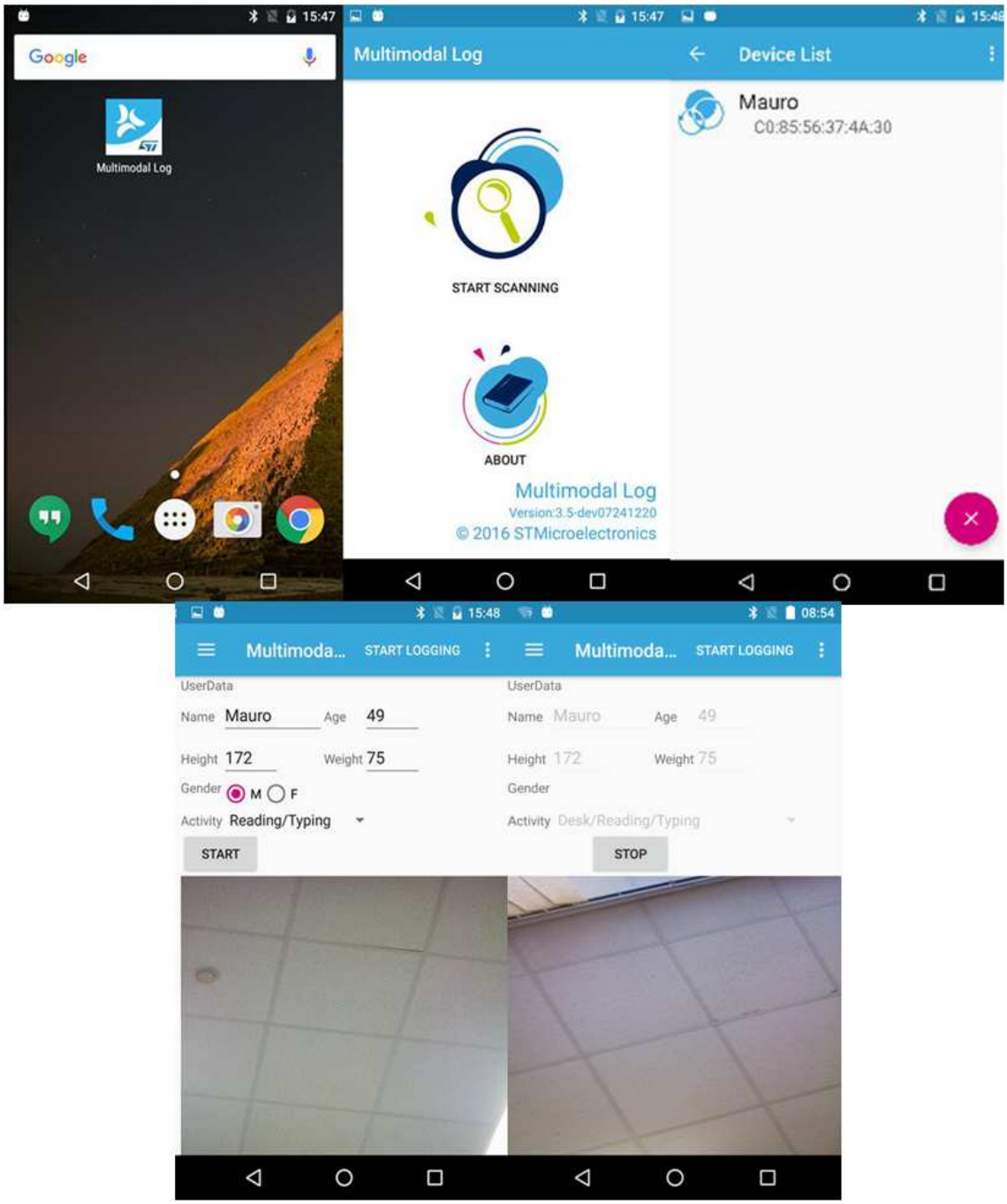


Figure 3.5: Smartphone app.

three axis from 5^{th} to 7^{th} column, magnetic field along three axis from 8^{th} to 10^{th} column and the last two columns list pressure and temperature data, respectively. Moreover, smartphone saves a *.mp4* file which contains the acquired video whereas

the information about the user and the time of the activity switch are saved in a *.txt* file. The name of each file comprises year (yyyy), month (MM), day (dd), hour (hh), minutes (mm) and seconds (ss).

In our case, the "Activity" section contains the following combination of transitions:

- Desk/Reading/Typing
- Desk/Typing/Reading
- Walking/Sitting/Desk/Standing
- Walking/Stairs.

In order to acquire all the transitions we defined, the last activity is the same as the first one. For example, if the selected transition is Walking/Stairs, the user collects the following sequence: Walking, Stairs and Walking again.
A double tap on the smartphone screen was used to registrer the activity change. The pressure of the tap is an important point, because if it is very strong, the collected video is affected by a sudden movement whereas if it is done "very gently", smartphone screen may not feel double tap and will not write the time of the activity change to the *.txt* file. "START", "STOP" and double tap are done with left hand because in the left wrist there is not Bluecoin sensor, otherwise, if one of these click are done with right hand, inertial data could be affected.

In some preliminary acquisitions, we noted that the BlueCoin sensors started recording at slightly different times, with respect to the smartphone, and consequently the data streams from the smartphone and BlueCoin were not perfectly synchronised. Audio and sensor data were collected with the same device (i.e. BlueCoin) therefore they were synchronised. At the beginning of each acquisition, each subject pronounced the word "START" in order to use the audio modality, that is common to both devices (smartphone and BlueCoin), to synchronize the signal. Some details of signals synchronization will be given in the subsection 3.4.

3.3 Dataset organization

This dataset was collected by 19 subjects and data are separated in 19 folders, one for each participant. In each of them, there are 4 folders which correspond to the acquired sequences and each of them contains the following data:

- *yyyyMMdd_hhmmss.txt*: user information, activity ground truth, transitions and timestamp
- *yyyyMMdd_ hhmmss.mp*4: video data
- *yyyyMMdd_ hhmmss.csv*: inertial data
- *yyyyMMdd_ hhmmss.wav*: audio data.

3.4 Signal Correlation

Cross-correlation is a technique used in signal processing which measures the similarity between two signals. So, given two signals $u(t)$ and $v(t)$, the cross-correlation between them is defined as follows:

$$\begin{aligned} w(t) = u(t) \otimes v(t) &= \int_{-\infty}^{\infty} \overline{u(t)}\, v(\tau + t) dt \\ &= \int_{-\infty}^{\infty} \overline{u(\tau - t)}\, v(t) dt \end{aligned} \tag{3.1}$$

where $\overline{u(t)}$ is the complex conjugate of signal u and τ is the time delay. The argument of $w(t)$ is called lag and it is the delay between the two signals. If $u(t) = v(t)$, the formula 3.1 defines the auto-correlation concept which is the cross-correlation of a signal with itself.

In our experiment, since the data collected with the BlueCoin board were aligned each other, the goal was the synchronization of the video with the Bluecoin data. To this aim, we extracted the audio signal from video sequences and calculated the cross-correlation between the audio signals captured by the smartphone and the BlueCoin board, respectively. The first three seconds of audio signals are considered because in this range of time the subjects says START. To this goal, the audio signals

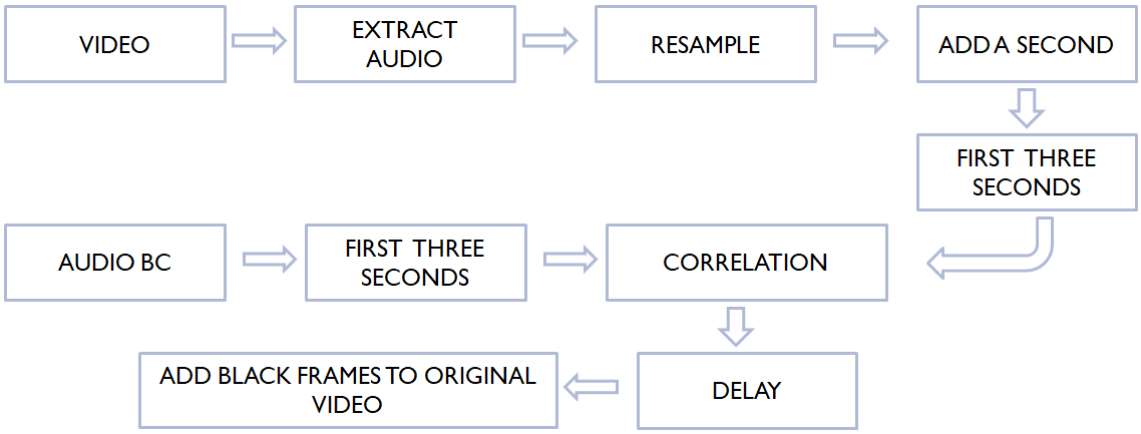


Figure 3.6: Align signal pipeline.

were extracted from each video in order to align the video data with Bluecoin data and only one channel was considered from audio of Bluecoin.

The correlation pipeline is synthetically sketched in Figure 3.6. Each audio signal extracted from videos was resampled on the original sampling rate 48KHz to the BlueCoin's audio sampling rate 32KHz. Since the word START is heard first in the audio extracted from the video sequence, the waveform related to this word is shifted a second forward. The cross-correlation of these signals and their lag are computed. Figure 3.7a) shows the audio signals which are extracted from an audio channel from the BlueCoin and from the corresponding video, a zoom of the first three seconds of these signals is given in 3.7b) whereas in Figure 3.7c) the first three seconds of audio signal from video, audio from Bluecoin and the aligned signals are shown, respectively.

The delay was expressed as number of samples (S). This is used to generate a zero S-dimensional vector that is concatenated to the audio signal extracted from video. The delay was also expressed in seconds (s) to align each video with the audio signal. To this aim, some black frames were placed at the beginning of the original video file. The number of frames (#F) added is given by the following equation

$$\#F = (1 + time) * fr, \tag{3.2}$$

where 1 is the second added at the beginning to shift START, time is the delay in seconds and fr is the video frame rate.

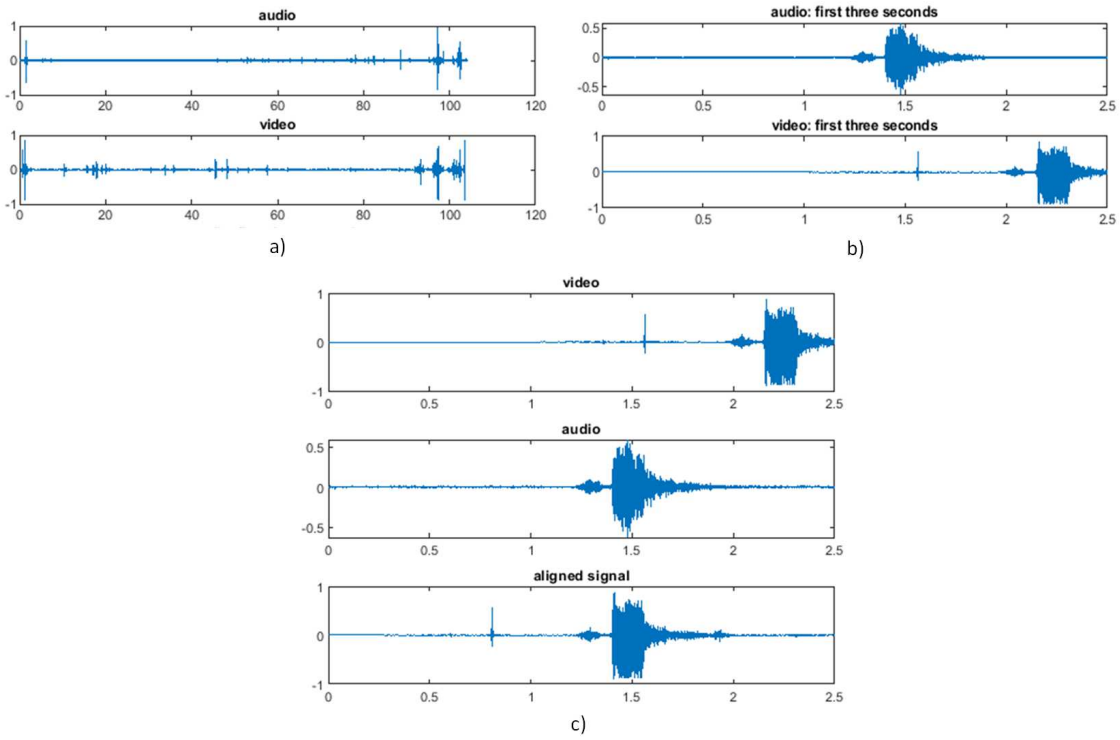


Figure 3.7: Correlation signal result.

3.5 Dataset Analysis

Removing the corrupted sequences, the final dataset is composed by includes 1325 multimodal data sequences each of which contains two or three transactions. Table 3.4 compares the relevant multimodal datasets available so far, by considering the main characteristics and the presence of transitions between activities. Our dataset (see next section for details) is included in the comparison. The second and third columns are referred to the acquisition modality. Fourth column indicates the number of classes, whereas column five and column six are related to the number of subjects involved into the acquisition and the number of sequences respectively. Also information about video frames resolution and the presence of transitions between consequent activities, and acquisition modalities are included. As it can be observed, our dataset has more sequences than the others. It has been specifically tailored for multimodal action anticipation. Most of the datasets present transitions between classes, but the number of transitions is not enough to train a neural network. Moreover, differently from the multimodal dataset presented in literature, ST

Activity	♯ of Sequances
Desk	2026
Reading	824
Typing	824
Walking	1353
Stairs	420
Sitting	405
Standing	405

Table 3.1: Number of sequences for each activity.

Past vs Fut	Desk	Reading	Typing	Walking	Stairs	Sitting	Standing
Desk		418	411				419
Reading	411		418				
Typing	418	411					
Walking					428	419	
Stairs				428			
Sitting	419						
Standing				419			

Table 3.2: Number of sequences for each transition. Rows indicate past whereas column future activities. Δ is equal to 1 second.

Multimodal Dataset comprises many different modalities, indeed, video, audio and sensor data (inertial and ambient sensor) are acquired.
Table 3.1 reports a summary of the number of sequences collected for each activity. "Desk" activity is more represented than "Stairs", "Sitting" or "Standing" activities because "Desk" is the past of "Typing", "Reading" and "Sitting" and the future of "Typing", "Reading" and "Standing".

Since the proposed dataset was collected for activity anticipation task, Table 3.2 shows a transition matrix and reports the number of sequences for each transition. For example, if the past is "Desk" and the Future is "Reading", we have 418 transitions. From "Reading" to "Desk", there are 411 sequences, and so on.

Table 3.2 also shows that the collected dataset is balanced, i.e. the same number of sequences for each transition was acquired. Sequences were divided in two part considering the transition point. We considered a $\Delta = 1$ before and after the transition point to obtain the data in Table 3.2. We also considered the case of $\Delta = 2$. When 2 seconds is considered, the transition matrix changes as shown in Table 3.3.

Past vs Fut	Desk	Reading	Typing	Walking	Stairs	Sitting	Standing
Desk		418	408				350
Reading	408		418				
Typing	418	408					
Walking					420	355	
Stairs				420			
Sitting	355						
Standing				412			

Table 3.3: Number of sequences for each transition. Rows indicate past whereas column future activities. Δ is equal to 2 seconds.

In this case, some transitions are lost because some activities, such as "Sitting" or "Standing", are shorten than considered Δ and are removed from the analysis.

3.6 Summary

In this chapter, ST Multimodal Dataset is introduced. It is a new multimodal dataset that we collected to anticipate the future activity observing only the past. Details about collection procedure and signal alignment have been given.

Dataset	First Person	Third Person	♯ Class	♯ Subjects	♯ Sequences	Resolution	Transition	Video	Audio	Inertial Sensors	Ambient Sensors
Multimodal Egocentric Activity Dataset [154]	✓	✗	20	-	200	1280x720	✗	✓	✗	✓	✗
Daily Intention Dataset [102]	✓	✗	34	3	164	640x480	✓	✓	✗	✓	✗
CMU-MMAC Dataset [149]	✓	✓	31	39	16	800x600	✓	✓	✓	✓	✗
Stanford-ECM Dataset [30]	✓	✗	24	10	113	720x1280	✓	✓	✗	✓	✗
Opportunity Dataset [151]	-	-	12	4	24	-	✓	✗	✗	✓	✓
UTD-MHAD Dataset [153]	✗	✓	22	8	861	640x480	✗	✓	✗	✓	✗
Multimodal User-Generated Videos Dataset [152]	✓	✗	24	1	19	1920x1080	✓	✓	✓	✓	✓
ST Multimodal Dataset	✓	✗	7	19	1325	720x1280	✓	✓	✓	✓	✓

Table 3.4: Relevant multimodal datasets together with main characteristics.

A	B	C	D	E	F	G	H	I	J	K	L
Timestamp (ms)	AccX (mg)	AccY (mg)	AccZ (mg)	GyroX (mdps)	GyroY (mdps)	GyroZ (mdps)	MagnX (mgauss)	MagnY (mgauss)	MagnZ (mgauss)	Pressure (hPA)	Temperatur (˚C)
0	-21	-845	-540	-2170	700	70	613	-913	-581	1012.4	37
19	-17	-834	-540	-1820	840	-140	609	-910	-589	1012.4	37
38	-17	-836	-537	-770	490	210	610	-916	-585	1012.3	37
57	-20	-851	-532	-630	-70	420	610	-916	-585	1012.3	37
76	-18	-842	-536	-1540	280	280	620	-912	-584	1012.4	37
95	-20	-844	-537	-1820	280	350	610	-916	-581	1012.4	37
114	-17	-840	-538	-1960	490	280	609	-913	-580	1012.3	37
133	-19	-843	-536	-1610	140	490	606	-919	-592	1012.3	37
152	-17	-842	-537	-1610	350	420	621	-919	-587	1012.3	37
171	-18	-842	-537	-1400	140	490	605	-923	-583	1012.3	37
190	-19	-843	-536	-1610	-70	350	613	-928	-587	1012.3	37
209	-18	-842	-539	-1820	0	280	609	-908	-589	1012.5	37
228	-18	-836	-538	-1540	-140	210	608	-930	-588	1012.4	37
247	-18	-836	-538	-1540	-140	210	605	-913	-587	1012.3	37
266	-20	-843	-538	-1260	-420	350	612	-927	-591	1012.3	37
285	-16	-838	-536	-1400	-420	280	610	-905	-593	1012.3	37
304	-14	-843	-536	-1330	-980	490	617	-912	-584	1012.3	37
323	-14	-840	-534	-1470	-1610	560	603	-917	-591	1012.3	37
342	-12	-839	-537	-1610	-2520	910	612	-923	-588	1012.5	37
361	-15	-840	-537	-1050	-3500	1400	616	-909	-607	1012.3	37
380	-20	-843	-535	-1400	-4550	1120	620	-929	-596	1012.4	37
399	-21	-840	-535	-1260	-5390	280	620	-913	-599	1012.4	37
418	-26	-842	-536	-1400	-5670	560	610	-924	-600	1012.4	37
437	-29	-841	-538	-1680	-5880	490	626	-918	-598	1012.3	37
456	-29	-837	-541	-1820	-5670	350	619	-915	-604	1012.4	37
475	-27	-843	-536	-1050	-5110	770	619	-915	-604	1012.4	37
494	-28	-847	-534	-1400	-5110	980	622	-933	-609	1012.3	37
513	-31	-842	-537	-2100	-4970	1120	616	-930	-607	1012.5	37
532	-32	-842	-536	-2170	-4830	1680	626	-933	-621	1012.3	37
551	-35	-842	-541	-2520	-4550	1470	615	-930	-621	1012.4	37.1
570	-36	-839	-540	-2030	-3990	1470	622	-932	-615	1012.4	37.1
589	-43	-846	-541	-1820	-3080	1750	622	-941	-621	1012.4	37
608	-39	-841	-541	-2520	-1890	1260	622	-959	-616	1012.3	37
627	-39	-841	-541	-2520	-1890	1260	616	-932	-626	1012.3	37
646	-35	-833	-543	-2030	-1540	1050	629	-939	-629	1012.4	37.1
665	-36	-836	-536	-1260	-1820	1540	619	-932	-625	1012.3	37.1
684	-38	-843	-540	-980	-2240	1540	620	-936	-630	1012.3	37.1
703	-41	-845	-537	-1470	-1820	1890	627	-940	-626	1012.3	37.1
722	-37	-840	-539	-2730	-1260	1890	636	-946	-633	1012.4	37
741	-40	-838	-542	-2310	-910	2310	625	-950	-629	1012.4	37.1
760	-39	-838	-538	-1960	-1190	2520	623	-944	-632	1012.4	37.1
779	-43	-840	-539	-1680	-1470	2380	622	-939	-624	1012.4	37.1
798	-48	-844	-541	-1890	-1330	2100	628	-953	-625	1012.3	37.1
817	-42	-842	-540	-3080	-700	1750	623	-950	-632	1012.4	37.1

Figure 3.8: Example of raw sensor data collected from Bluecoin.

Chapter 4

Action Anticipation from Multimodal Data

The prediction of the future is a challenge that has always fascinated humans. The overall design of machines that anticipate future actions is still an open issue in Computer Vision. In this chapter, the problem of predicting user actions is considered. To our knowledge, most papers on action anticipation consider only video data, this chapter presents a study of predicting a future action from currently observed multimodal data. In particular, our goal is to build a shared representation related to data coming from different domains, such as images, audio signal, heart rate, acceleration, etc., in order to anticipate daily activities of a user wearing multimodal sensors. To this aim, in Section 4.1, we consider the Stanford-ECM Dataset [30] which comprises video, acceleration and heart rate data. The dataset is adapted to our action prediction task, by identifying the transitions from the generic "Unknown" class to a specific "Activity". Auto-encoder and Siamese network with Multi Layer Perceptron and 1D CNN are used for predicting next activity just from features extracted from the previous temporal sequence, labelled as "Unknown".

The used dataset is not specific for action anticipation task. Moreover, the transitions from a generic unknown class to a specific activity had few sequences and the adapted dataset was strongly unbalanced. To overcome these difficulties, in the Section 4.3, the pipeline, presented in 4.1, is considered and it is evaluate on ST Multimodal Dataset, introduced in Chapter 3. Our goal is to provide proof that our pipeline is able to anticipate future action from an observed past clips, given more past sequences.

The prediction accuracy of the tested models is compared with respect to the

classic action classification which is considered as a baseline. Results demonstrate that the presented system is effective in predicting activity from a past observation and suggest that multimodality improves both classification and prediction in some cases. This confirms that data from different sensors can be exploited to enhance the representation of the surrounding context, similarly to what happens for human beings, that elaborate information coming from their eyes, ears, skin, etc. to have a global and more reliable view of the surrounding world.

	Bicy-cling Uphill	Runnin g	Bicy-cling	Playing With Childre n	Resi-stance Trainin g	Ascen-ding Stairs	Cali-sthe-nics	Wal-king	Desce n-ding Stairs	Coo - king	Presen -ting	Drivin g	Shop -ping	Strol-ling	Food Prepa-ration	Talkin g Stan-ding	Talkin g Sitting	Sitting Tasks	Mee-ting	Eating	Stan-ding In Line	Riding	Rea-ding	Back-groun d	Unknow n
Bicyclin g Uphill			2																						13
Running								1																	4
Bicyclin g	3																							1	23
Playing With Children																						1			42
Resistan-ce Training																									6
Ascen-ding Stairs								5																	15
Calisthe-nics																									3
Walking						8			8					1	1	1					2				107
Descen-ding Stairs								8					1												13
Cooking															2										13
Presen-ting																									8
Driving																									7
Shop-ping									1																19
Strolling				1		1									1	1							2	1	45
Food Prepara-tion										5											1				18
Talking Standing																					1				42
Talking Sitting																				2					26
Sitting Tasks																				8					24
Meeting																									11
Eating																	2	8							10
Standing In Line													5			1									6
Riding				1																				1	8
Reading														1											9
Back-ground																						1			45
Unknow n	12	5	25	40	6	11	3	113	13	9	6	7	14	50	20	40	25	23	11	10	8	8	8	43	

Figure 4.1: Transition matrix: Past vs Future.

Figure 4.2: Cut of the dataset: 64 sample before and after transition point are considered.

4.1 Proposed approach on Stanford ECM Dataset

In this section, our approach that is able to anticipate an activity from multimodal signals is introduced. To this aim, Stanford-ECM Dataset [30] is considered. In the following, dataset adaptation and the building blocks of our system are detailed.

Stanford-ECM Dataset [30] was created for classification task and it is described in Chapter 3. It was necessary to adapt it to our action anticipation task. At first time, the transition matrix, which is shown in Figure 4.1, was studied. The first dark blue column represents the past whereas the first dark blue row is the activity of the future. It is easy to note that the matrix is sparse, this means that there are not enough transitions from a specific activity to another. For example, there are only 8 transitions from "Walking" to "Descending Stairs" or only one transition from "Running" to "Walking". On the other hand, it is clear that most of the transitions come from a generic "Unknown" class to a specific activity (last dark blue row) or from a specific activity to "Unknown", for example, we have 113 clips from "Unknown" to "Walking" and 107 from "Walking" to "Unknown".

In this study, we considered a transition, suitable to build training and test sets: Unknown/Activity, where "Activity" means a generic activity different from "background" and "unknown". As shown in Figure 4.2, each modality is cut around the Unknown/Activity transitions including 64 samples (or frames for video) before and 64 after the transitions point which is the point where the activity changes.

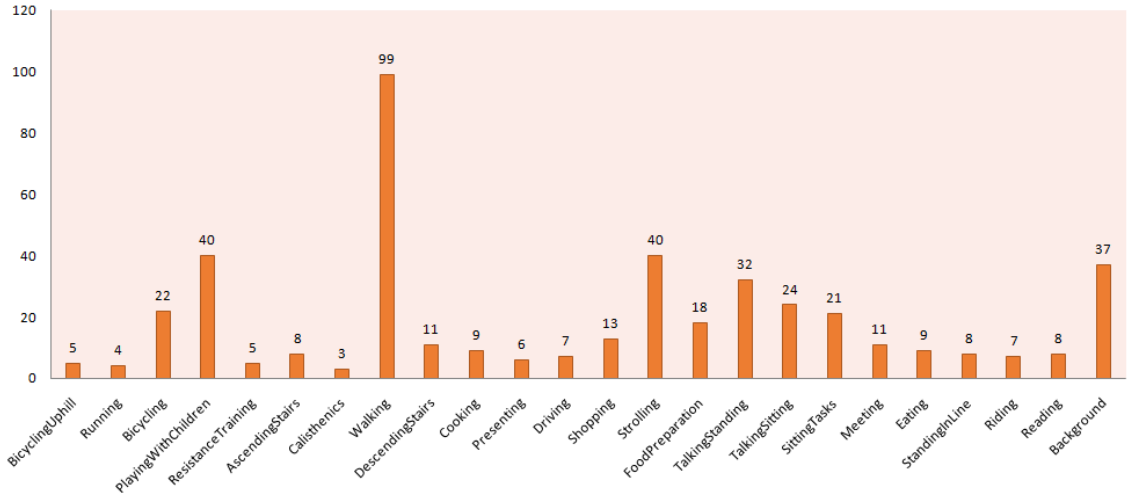


Figure 4.3: Number of Unknown/Activity transitions for each activity considered in this paper.

Figure 4.3 shows the number of segments of the Unknown/Activity transitions remaining after this process. Since some transitions were represented with few samples, we have concentrated the analysis to the following 9 activities: Bicycling, Playing With Children, Walking, Strolling, Food Preparation, Talking Standing, Talking Sitting, Sitting Tasks and Shopping. Hence, the final dataset contains 309 transitions Unknown/Activity.

Our problem is defined as follows. Let be $\mathbf{y}_t = (\mathbf{v}_t, \mathbf{a}_t, hr_t)^T$ the input vector at time t where $\mathbf{v}_t \in \mathbb{R}^2$ is a video, $\mathbf{a}_t \in \mathbb{R}^3$ is an acceleration signal and $hr_t \in \mathbb{R}$ is a heart rate data, we define the feature representation of video, acceleration and heart rate signal as $\mathbf{x}_t^v$, $\mathbf{x}_t^a$ and $\mathbf{x}_t^{hr}$ and $\mathbf{x}_t = (\mathbf{x}_t^v, \mathbf{x}_t^a, \mathbf{x}_t^{hr})^T$ the features vector at time t. Given $\mathbf{x}_t$ as input, we want to predict the label $label_{t+1}$ of the next action by observing only data before the activity starts.

4.1.1 Proposed Approach

The proposed approach is synthetically sketched in Figure 4.4. The model considers three modalities video, acceleration and heart rate as input after a feature extraction process. Moreover details of the different components of our approach will be given.

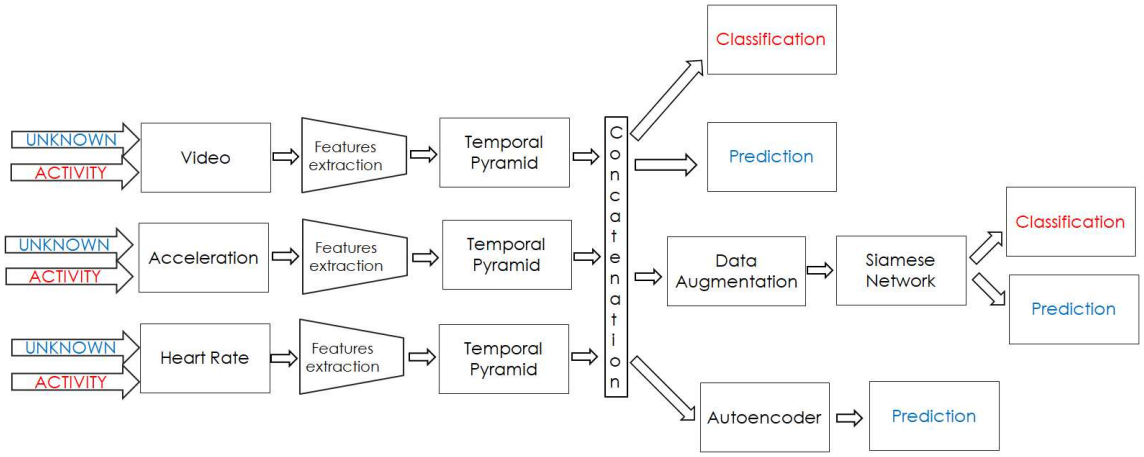


Figure 4.4: Pipeline of our anticipation approach.

Features Extraction

In this section we describe the feature representation $\mathbf{x}_t^v$, $\mathbf{x}_t^a$ and $\mathbf{x}_t^{hr}$ for each signal. The extraction of video and acceleration features is similar to [30].

For visual data, features are extracted from the pooling layer five of the Inception CNN architecture [15] pretrained on ImageNet [156]. Each video frame has been transformed into a $\mathbf{x}_t^v$ feature vector of 1024 dimension.

To reduce the computational complexity and redundancy, for sensor data (acceleration and heart rate data), we decided to extract features from the raw sensor measurements, that characterize the original signals, by considering time and frequency domains rather than use a neural network. For acceleration data, we extracted features from raw signals through a temporal sliding window process considering a window size of 32fps. Time-domain features are computed from the time-series of the original data, whereas frequency-domain features are extracted from the spectral analysis of the signal. The values of spectral coefficient represent the magnitude of frequency component [157]. For time-domain features, mean, standard deviation, skewness, kurtosis, percentiles (10th, 25th, 50th, 75th, 90th), acceleration count for each axis and correlation coefficients between each axis are computed. For frequency-domain features, we consider the spectral entropy

$$J = -\sum_{i=0}^{N/2} \bar{P}_i \cdot \log_2 \bar{P}_i$$

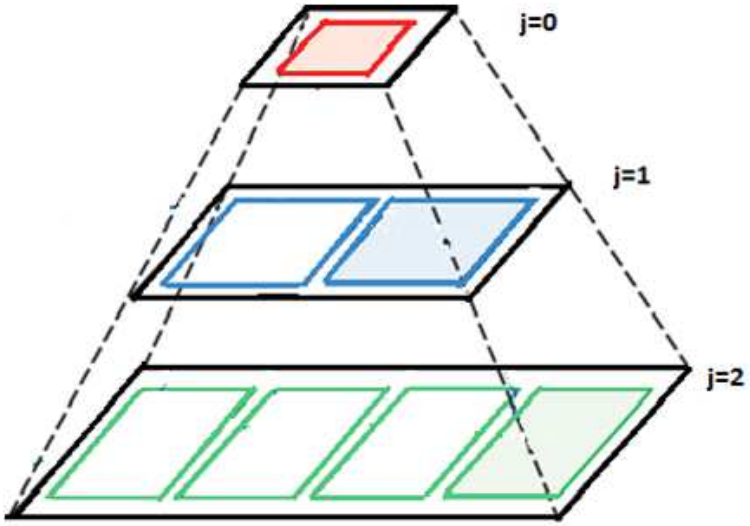


Figure 4.5: Temporal Pyramid.

where $\bar{P}_i$ is the normalized power spectral density computed from Short Time Fourier Transform (STFT), which measures the distribution of frequency component normalized by the window. Then, the obtained features from these domains are concatenated and $\mathbf{x}_t^a$ is a 36-dimensional vector. For heart rate data, the features are extracted from the time-series of the raw signals. Mean and standard deviation are calculated to compute a $\mathbf{x}_t^{hr} \in \mathbb{R}^2$ vector.

Temporal Pyramid

We represent features in a temporal pyramid fashion [158] composed by three level, as shown in Figure 4.5.

The top level (j=0) is an histogram over the full temporal extent of a data, the next level (j=1) is the concatenation of two histograms obtained by temporally segmenting each modality into two halfs, the last level (j=2) is the concatenation of four histograms obtained by temporally segmenting each previous histogram into two halves.

In this way, 7 histograms are obtained corresponding to a 1024×7 visual features, 36×7 acceleration features, and 2×7 heart rate features. All features are concatenated into a single vector $\mathbf{x}_t = (\mathbf{x}_t^v, \mathbf{x}_t^a, \mathbf{x}_t^{hr})^T$ of 7434 components.

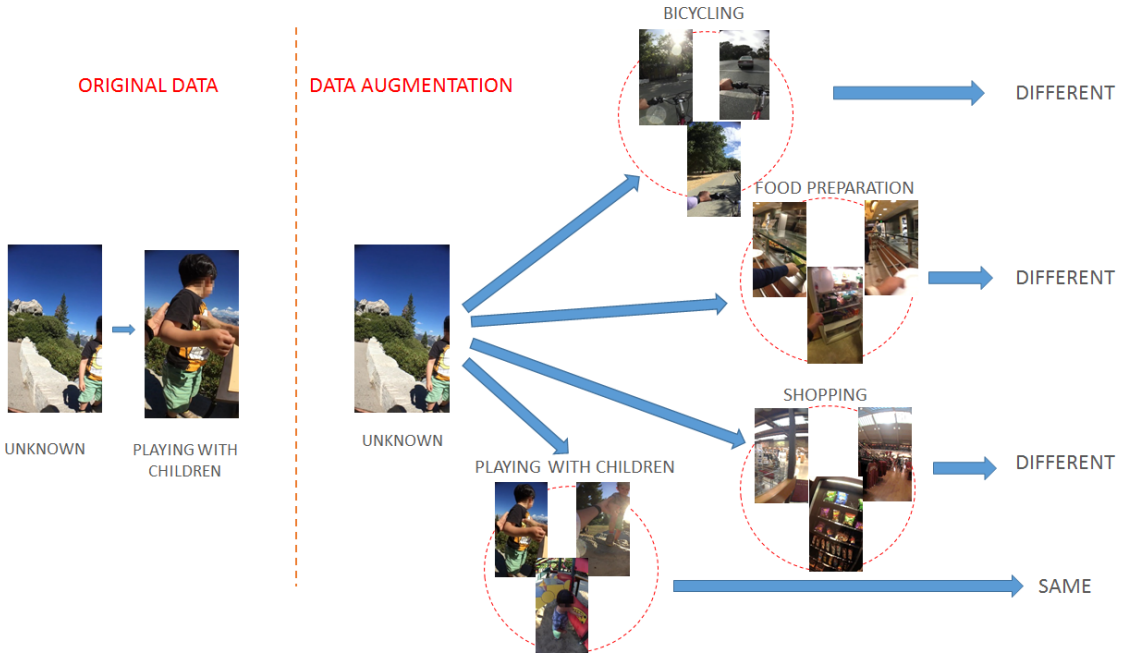


Figure 4.6: Data Augmentation for Siamese Network.

Data Augmentation

Since we have few transition samples, data augmentation technique is used to expand the training set to prevent over-fitting. As reported in [14], geometric transformation and RGB channels alteration are the traditional data augmentation approaches which can be applied to images, while [159] proposes a method which exploit Generative Adversarial Network (GAN) and CycleGAN.

The permutation Unknown/Activity is considered, therefore each unknown sequence is paired with all the possible sequences of activity. The label of each augmented transition is changed from 0-8 to 0-1, as follows:

$$label = \begin{cases} 1 & \text{if } l_o = l_a \\ 0 & \text{otherwise} \end{cases}$$

where l_o is the label of the activity in original dataset while l_a is the label in augmented dataset. An example is shown in Figure 4.6. In other words, if unknown and the activity belong to the same class (e.g. unknown related to playing with children and the following activity is playing with children), we assign a label 1, otherwise a

label 0 is assigned if unknown and the activity are different (e.g. unknown related to playing with children and activity is related to food preparation).

The obtained dataset is strongly unbalanced. Table 4.1 compares the number of sequences before and after augmentation. Some classes, such as Shopping or Food Preparation, are poorly represented therefore it is necessary to down-sample the dataset.

Activity	♯ of original activity transitions	♯ of augmented activity transitions	♯ of final transitions
Bicycling	18	4482	1353
Walking	79	19671	1353
Shopping	11	2739	1353
Talking Standing	26	6474	1353
Sitting Tasks	17	4233	1353
Playing With Children	32	7968	1353
Strolling	32	7968	1353
Food Preparation	14	3486	1353
Talking Sitting	20	4980	1353
TOT	249	62001	12177

Table 4.1: Number of sequences for each activity before and after augmentation.

We consider the square of minimum value of the number of original activity transitions ($11^2 = 121$) from sequences with label 1 and 154 sequences from sequences with label 0 for each class, in order to balance activities classes and unknown class. The final dataset has 12177 sequences.

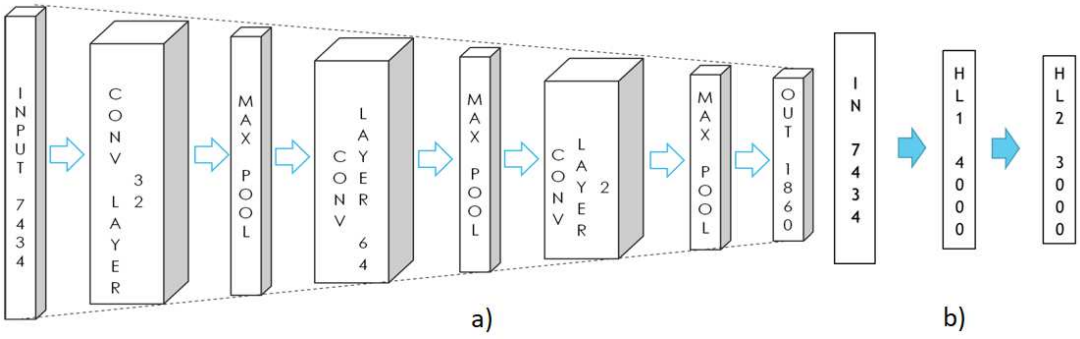


Figure 4.7: a) 1D CNN Architecture. b) MultiLayer Perceptron Architecture.

Learning Approach

Our goal is to build an embedding space where the unknown sequences, which are related to the past, are close to those of future activities. In this regard, we use Auto-encoders and Siamese Networks.

Since we want to use an Auto-encoder to predict the future activity from a generic unknown past, it is trained like a denoising auto-encoder where the input is an unknown clip and the output is an activity sequence. For example, if the transition is "Unknown" "Playing with Children", the past clip ("Unknown") is given to the network as input and the network is trained to reconstruct the future activity ("Playing with Children"). To measure the probability error, cross- entropy loss is computed:

$$\mathcal{L}(y) = -(y \log(p(y)) + (1 - y) \log(1 - p(y))) \tag{4.1}$$

where y is the ground truth activity label and $p(y)$ is the predicted probability.

Siamese network is also trained to make representations of features of "Unknown" sequences and next "Activity" very close in the embedding space. One stream of the Siamese network processes the unknown features whereas the other stream processes those related to the activity. Euclidean metric is used as distance between inputs. The contrastive loss function [42] is used for training purposes:

$$\mathcal{L}(y) = y\sqrt{D} + (1 - y)\sqrt{\max(1 - D, 0)} \tag{4.2}$$

where y is the ground truth activity label and D is the euclidean distance between two feature points.

We consider two different architectures for Auto-encoder and Siamese network : Multilayer Perceptron [62] and 1D Convolutional Neural Network (CNN) [160, 161]. Figures 4.7 shows the architecture of the used networks. For Multilayer Perceptron, two hidden layers are considered with a number of neurons of 4000 and 3000 respectively. For 1D CNN, three convolutional layers are used with a number of filters equal to 32, 64 and 2, respectively, (all of size 3×1) and a relu activation function. The output of each convolutional layer is reduced in size using a max-pooling layer that halves the number of features.

Classification and Prediction

Our aim is to predict next activity from an unknown clip. To our knowledge, in the state of the art, there are not results on action anticipation from multimodal data, therefore we consider as baseline the classification of activity sequences and the

classification of unknown sequences. A k-nearest-neighbor classification algorithm (K-NN) and a support vector machine (SVM) are used for classification purposes.

4.2 Experimental Results on Stanford ECM Dataset

In this section, the results of the proposed approach are shown and discussed. Our model is evaluated on the Stanford-ECM Dataset. The feature representations obtained with the considered deep architectures are classified with SVM or K-NN classifier.

4.2.1 Setup

We randomly split our data into disjoint training (249 sequences) and testing sets (60 sequences) for training and testing purposes. Auto-encoder is trained for 10000 epochs and the optimizer is Adam. Learning rate decreases in exponential fashion and it starts from 0.003. For Siamese Network, the Adam optimizer is considered with batch size of 249 samples. Variable learning rate is used starting from 0.001. In the Multilayer Perceptron, in order to prevent overfitting, we apply a dropout procedure during training. We evaluate K-NN for different values of k and SVM for different kernels. In K-NN classifier, we consider two different weights: uniform and distance. The first assigns equal weights to all points, while distance weight assigns weights proportional to the inverse of the distance from the query point.

4.2.2 Baseline

In order to better evaluate our approach, we define a baseline where the values of accuracy in classification and in prediction are compared. In classification, the features related to activity sequence, extracted as described in session 4.1.1, are classified, while in prediction we consider the classification of features related to unknown clips.

The Tables 4.2 and 4.3 show the values of accuracy for each signals and combinations of all of them. For example, if we consider the accuracy values of video features, in Table 4.2, we can see that, with a linear kernel, we obtain an accuracy

	Classification		**Prediction**	
Modality (♯ Features)	**Linear Kernel**	**RBF**	**Linear Kernel**	**RBF**
Acceleration (252)	31.67%	46.67%	31.67%	46.67%
Heart rate (14)	33.33%	28.33%	33.33%	35%
Video(7168)	66.67%	68.33%	60%	56.67%
Acceleration+Heart rate (266)	36.67%	50%	38.33%	48.33%
Video+Acceleration (7420)	70%	71.67%	68.33%	68.33%
Video+Heart rate(7182)	66.67%	66.67%	60%	63.33%
Video+Acceleration+Heart rate (7434)	70%	68.33%	68.33%	68.33%

Table 4.2: Baseline Results: SVM.

	Classification									
Modality (♯ Features)	**weights=uniform**					**weights= distance**				
	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
Acceleration (252)	41.67%	53.33%	48.33%	45%	45%	41.67%	51.67%	46.67%	50%	50%
Heart rate (14)	23.33%	21.67%	18.33%	23.33%	31.67%	23.33%	20%	15%	15%	15%
Video(7168)	63.33%	61.67%	61.67%	61.67%	60%	63.33%	61.67%	63.33%	66.67%	63.33%
Acceleration+Heart rate (266)	38.33%	46.67%	48.33%	45%	43.33%	38.33%	43.33%	45%	48.33%	46.67%
Video+Acceleration (7420)	61.67%	61.67%	63.33%	63.33%	61.67%	61.67%	58.33%	61.67%	65%	65%
Video+Heart rate(7182)	63.33%	61.67%	61.67%	61.67%	60%	63.33%	61.67%	65%	65%	63.33%
Video+Acceleration+Heart rate (7434)	60%	65%	63.33%	63.33%	58.33%	60%	61.67%	63.33%	65%	63.33%
	Prediction									
Modality (♯ Features)	**weights=uniform**					**weights= distance**				
	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
Acceleration (252)	41.67%	53.33%	48.33%	45%	45%	41.67%	51.67%	46.67%	50%	50%
Heart rate (14)	20%	26.67%	26.67%	25%	35%	20%	20%	20%	23.33%	26.67%
Video(7168)	55%	56.67%	60%	56.67%	60%	55%	58.33%	58.33%	56.67%	63.33%
Acceleration+Heart rate (266)	33.33%	46.67%	48.33%	45%	48.33%	33.33%	41.67%	45%	45%	48.33%
Video+Acceleration (7420)	53.33%	58.33%	60%	60%	56.67%	53.33%	61.67%	58.33%	60%	60%
Video+Heart rate(7182)	55%	56.67%	60%	56.67%	60%	55%	58.33%	58.33%	56.67%	63.33%
Video+Acceleration+Heart rate (7434)	53.33%	58.33%	61.67%	60%	56.67%	53.33%	61.67%	60%	60%	58.33%

Table 4.3: Baseline Results: K-NN.

value of 66.67% in classification and a value of 60% in prediction; if we combine video features with acceleration data, for instance, the values are 70% in classification and 68.33% in prediction. These results suggest two conclusions. The first is that, as it is easily understandable, the values of accuracy in classification are higher than those in prediction, but not so much higher, therefore it is possible to anticipate the future action. The second is that most of the information comes from the video, but if we combine video with another signal, such as acceleration, the value of accuracy increases. The same conclusions are obtained with K-NN classifier.

4.2.3 Auto-encoder and Siamese Network

Our goal is to predict the label of the next action by observing only data before the activity starts. Our baseline suggests that it is necessary to fill the gap between

(a)

KNN									
weights=uniform					**weights=distance**				
$K=1$	$K=3$	$K=5$	$K=7$	$K=9$	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$
Baseline: Classification									
60%	65%	63.33%	63.33%	58.33%	60%	61.67%	63.33%	65%	63.33%
Baseline: Prediction									
53.33%	58.33%	61.67%	60%	56.67%	53.33%	61.67%	60%	60%	58.33%
Auto-encoder									
56.67%	55%	55%	55%	60%	56.67%	53.33%	55%	56.67%	60%

(b)

SVM	
Linear Kernel	**RBF Kernel**
Baseline: Classification	
70%	68.33%
Baseline: Prediction	
68.33%	68.33%
Auto-encoder	
51.67%	63.33%

Table 4.4: Auto-encoder Results considering a MultiLayer Perceptron architecture.

(a)

	KNN									
	weights=uniform					**weights=distance**				
♯ of hidden layers(♯ of neurons)	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$
1 (3000)	58.33%	53.33%	50%	48.33%	48.33%	58.33%	55%	51.67%	51.67%	51.67%
3 (3000)	38.33%	45%	50%	48.33%	41.67%	38.33%	45%	48.33%	50%	48.33%

(b)

	SVM	
♯ of hidden layers(♯ of neurons)	**Linear Kernel**	**RBF Kernel**
1 (3000)	58.33%	61.67%
3 (3000)	50%	50%

Table 4.5: Auto-encoder results considering a MultiLayer Perceptron architecture with one hidden layer and three hidden layers.

the accuracy of classification and that of the prediction. As discussed in previous section 4.1.1, we test an Auto-encoder and a Siamese network for our purpose. Two different architectures are used for each network: Multilayer Perceptron (MLP) and a 1D CNN. The interesting point is that with a 1D CNN we can consider three

(a)

KNN									
weights=uniform					**weights=distance**				
$K=1$	$K=3$	$K=5$	$K=7$	$K=9$	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$
Baseline: Classification									
60%	65%	63.33%	63.33%	58.33%	60%	61.67%	63.33%	65%	63.33%
Baseline: Prediction									
53.33%	58.33%	61.67%	60%	56.67%	53.33%	61.67%	60%	60%	58.33%
Auto-encoder									
53.33%	48.33%	50%	50%	58.33%	53.33%	53.33%	53.33%	53.33%	53.33%

(b)

SVM	
Linear Kernel	**RBF Kernel**
Baseline: Classification	
70%	68.33%
Baseline: Prediction	
68.33%	68.33%
Auto-encoder	
61.67%	65%

Table 4.6: Auto-encoder Results considering a 1D CNN architecture.

convolutional layers therefore our output has dimension of 1860 while with a MLP we have only two layers and the output size is 3000.

Auto-encoder. Results of Auto-encoder are listed in the Table 4.4 and Table 4.6, in particular K-NN performance are shown in Table 4.3(a) and in Table 4.5(a) whereas Table 4.3(b) and Table 4.5(b) exhibit the SVM results. Since our goal is to fill the gap between the values of accuracy in classification and those in prediction, the tables compare the performances of each classifier in classification and prediction with the accuracy in prediction obtained with Auto-encoder. Considering both architectures MLP and 1D CNN, results do not reach the accuracy of the baseline because the network computes many parameters and the dataset is very small, indeed the auto-encoder is trained with 249 samples. However, this is not true for K-NN with $K = 1$ in Table 4.3(a). Indeed, in this case, we have 60% of accuracy in classification, 53.33% in prediction whereas with an auto-encoder, accuracy of 56.67% is obtained. In other words, the network overcomes the baseline accuracy value in prediction. Otherwise, in Table 4.5(a), auto-encoder gives an accuracy that

is equal to the performance in prediction (53.33%).

To simplify the auto-encoder with a MLP, only one hidden layer with 3000 neurons is considered. The performance are lower than the previous one, except for K-NN with $K = 1$, as it can be seen in Table 4.4(a). This could be due to the strong compression of the input data, in fact input has dimension 7434 and with only one layer, it is compressed into a 3000-dimensional vector.

To improve our results, auto-encoder with MLP has been made deeper than the previous one with two hidden layers, indeed three hidden layers are considered with a number of neurons of 5000, 4000, 3000 respectively. The performances are below the baseline results; for instance, for $K = 1$ (Table 4.4(a)), the accuracy is 38.33%, which means 20% less than previous network. With one layer and three layers, SVM does not achieve better results than baseline accuracy (Table 4.4(b)).

Since the results with an Auto-encoder are not what we expected, Siamese Network is considered to our task.

Siamese Network. Table 4.7 and Table 4.8 show the results of the Siamese network. The tables list the obtained accuracy with K-NN and SVM classifier both for classification and anticipation. With a Siamese Network composed by a Multilayer Perceptron, results on anticipation are not so good and are even worse, in most cases, than those obtained by the baseline. This could be due to the difficulty of the MLP to learn from a very tiny dataset. More in details, the number of parameters of the network (7434x4000+4000x3000) is too big with respect to the dataset size. Table 4.8 shows results obtained by training the considered classifiers on the representation learned through a Siamese Network, by exploiting a 1D convolutional layer architecture.

The best values of accuracy are obtained with K-NN for $K = 5$ and $K = 7$. Indeed, if we compare the accuracy values of our baseline in the Table 4.8 for $k = 5$ and weights=distance, we have 63.33% for classification, 60% for prediction whereas the Siamese network overcomes these values obtaining a 66.67% of accuracy. For k=7, results show that the accuracy value with a Siamese network is equal to 65%, which is the same value of accuracy obtained for classification baseline. It is also interesting to note that the representation generated by the Siamese Network is not suitable in this case for classification task; in fact, accuracy achieved in classification is quite lower than that of the simple baseline. This could be due to the fact that the

	Classification		**Prediction**	
	Baseline	**Siamese**	**Baseline**	**Siamese**
K	**KNN - weights=uniform**			
1	60%	58.33%	53.33%	55%
3	65%	60%	58.33%	55%
5	63.33%	58.33%	61.67%	55%
7	63.33%	56.67%	60%	53.33%
9	58.33%	56.67%	56.67%	53.33%
K	**KNN - weights= distance**			
1	60%	58.33%	53.33%	55%
3	61.67%	60%	61.67%	55%
5	63.33%	58.33%	60%	55%
7	65%	56.67%	60%	53.33%
9	63.33%	56.67%	58.33 %	53.33%
	SVM - Linear Kernel			
	70%	58.33%	68.33%	55%
	SVM - RBF Kernel			
	68.33%	46.67%	68.33%	56.67%

Table 4.7: Siamese Network Results considering a MultiLayer Perceptron architecture.

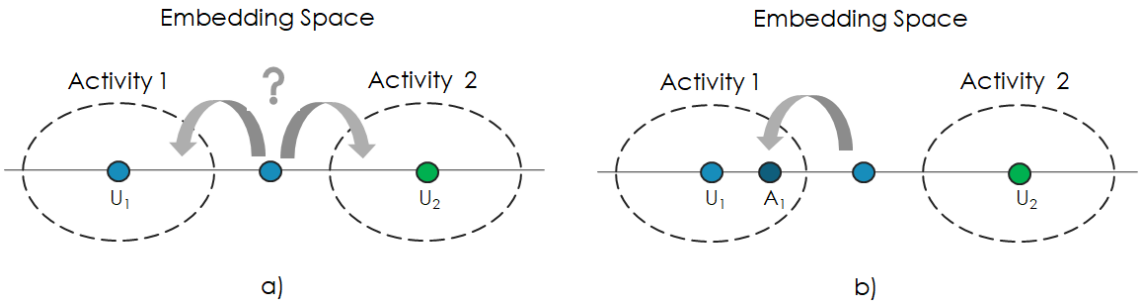


Figure 4.8: Projection of unknown features and activity features to improve the performance of K-NN.

Siamese network has been trained to solve the challenge of making representations of features of "Unknown" sequence and next "Activity" very close in the embedding space with few samples. The results achieved with the SVM classifier do not reach the accuracy of the baseline.

Figure 4.8 shows how is possible to improve the K-NN results. Figure 4.8a) illustrates a possible situation in an embedding space. Given two different activities,

	Classification		**Prediction**	
	Baseline	**Siamese**	**Baseline**	**Siamese**
K	**KNN - weights=uniform**			
1	60%	50%	53.33%	55%
3	65%	50%	58.33%	51.67%
5	63.33%	55%	61.67%	63.33%
7	63.33%	55%	60%	63.33%
9	58.33%	55%	56.67%	58.33%
K	**KNN - weights= distance**			
1	60%	50%	53.33%	55%
3	61.67%	53.33%	61.67%	58.33%
5	63.33%	55%	60%	66.67%
7	65%	58.33%	60%	65%
9	63.33%	60%	58.33 %	65%
	SVM - Linear Kernel			
	70%	71.67%	68.33%	60%
	SVM - RBF Kernel			
	68.33%	65%	68.33%	60%

Table 4.8: Siamese Network Results considering a 1D CNN architecture.

if another element is projected in the middle of them, a wrong classification could be done. So, projecting also the future activities in this embedding space, the classifier has many information to guess the true label, as shown in Figure 4.8b). The results obtained with Siamese Network considering a 1D CNN architecture are listed in the Table 4.9. As it can be observed, in some cases the results are better than the previous one. For instance, for $K = 3$ and weight=distance, we obtain 63.33% whereas in Table 4.8 we have 58.33%. Therefore in this way it is possible to improve results with respect to those exhibited in Table 4.8.

4.3 Proposed approach on ST Multimodal Dataset

The pipeline used in this paper is similar to the one presented in the previous Section. Differently than Section 4.1.1, in this Section we consider inputs of more sources: video, audio and sensor data, also to improve the performances, we considered a Triplet Network architecture [43]. The pipeline is sketched in the Figure 4.9.

	KNN			
	Classification		Prediction	
	weights=uniform	**weights=distance**	**weights=uniform**	**weights= distance**
1	51.67%	51.67%	60%	60%
3	55%	55%	63.33%	63.33%
5	50%	56.67%	61.67%	63.33%
7	51.67%	55%	61.67%	65%
9	53.33%	55%	60%	65%

Table 4.9: Siamese Network with 1D CNN architecture. Possible way to improve K-NN results.

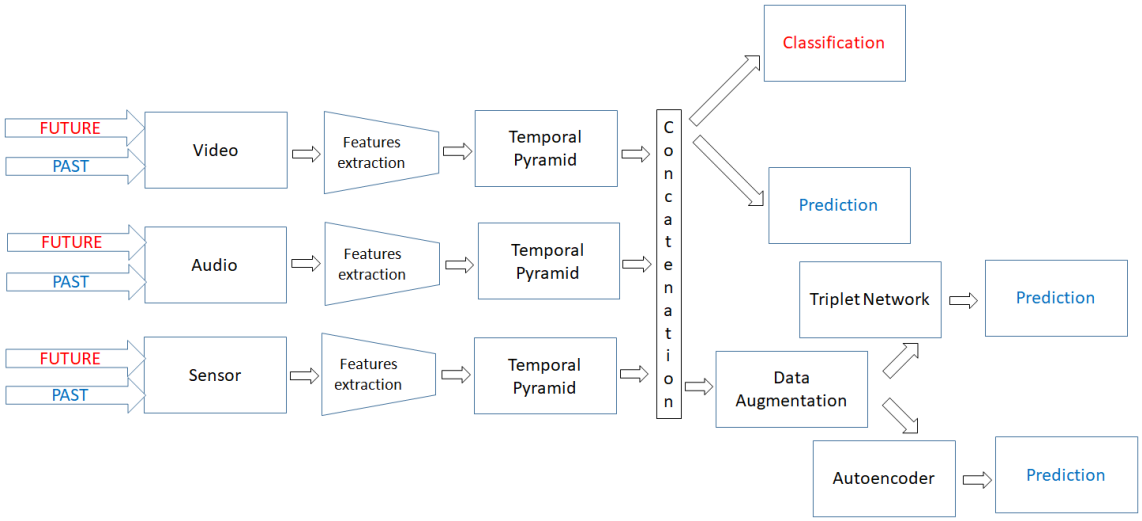


Figure 4.9: Proposed approach with ST Multimodal Dataset.

Each input is cut around the transition from a past action to a future activity including just over a second. Since each modality has different sampling rate, the number of samples for each of them is the following:

- Video: 36 frames;
- Audio: 32768 samples;
- Sensor: 64 samples.

4.3.1 Audio features extraction

Audio is converted into a vector which contains all the main information about the signal. There are many studies about audio and its application on speech recognition[162, 163], action recognition [113, 114] and video captioning [80]. Audio

♯ of Split)	Classification					Anticipation				
	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
1	22.23%	22.85%	22.54%	24.18%	24.28%	21.82%	22.95%	23.87%	25%	25.61%
2	19.67%	20.49%	21.11%	21.82%	22.24%	22.03%	21.62%	23.05%	22.34%	23.16%
3	22.95%	23.26%	23.16%	23.77%	23.77%	21.41%	22.85%	23.36%	23.16%	24.69%
4	21.52%	21.52%	22.95%	22.23%	23.46%	24.39%	24.90%	24.59%	24.69%	24.18%
5	21.11%	21%	21%	22.03%	22.85%	22.85%	25.31%	26.23%	26.95%	28.59%
Avg	21.50%	21.82%	22.15%	22.81%	23.32%	22.50%	23.53%	24.22%	24.43%	25.25%

Table 4.10: K-NN Results on ZCR and RMSE Features.

Features	**K-NN**					**SVM**	
	$K=1$	$K=3$	$K=5$	$K=7$	$K=9$	linear	rbf
Spectrogram	53.17%	55.16%	56.35%	54.76%	52.38%	52.78%	55.95%
MFCC	60.32%	62.30%	63.09%	63.09%	61.11%	62.30%	65.87%
MFCC (No T.P.)	55.56%	56.35%	57.14%	58.33%	57.93%	51.19%	66.27%
ZCR+RMSE	47.22%	48.81%	51.19%	51.19%	49.20%	48.81%	57.14%
ZCR+RMSE+MFCC	59.13%	60.71%	62.70%	61.90%	61.51%	63.10%	65.87%

Table 4.11: Classification results considering only sequences related to the classes: "Typing", "Walking", "Stairs".

Past/Fut	Desk	Reading	Typing	Walking	Sitting	Standing	Stairs
Desk	151	31	36	14	4	4	3
Reading	100	24	21	13	1	2	1
Typing	99	8	42	11	1	1	1
Walking	67	9	5	56	2	1	23
Sitting	50	8	9	6	3	1	3
Standing	59	4	6	4	3	2	3
Stairs	11	0	4	43	0	0	26

Table 4.12: Classification on the ZCR and RMSE audio features.

features extraction is based on different techniques and the main ways to pull out information from audio are:

- Temporal and Spectral Features;
- Mel-Frequency Cepstral coefficient (MFCC) [17];
- Spectrogram.

Temporal features are computed from the waveform of the signal or its energy. An example of these features is given by zero-crossing rate, which counts the number of times the signal crosses the horizontal axis, or by the root mean square energy, which compute the root-mean-square energy of audio signal. Spectral features are

2 seconds										
	Classification					**Prediction**				
♯ of Split	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$
1	37.28%	37.28%	38.22%	39.06%	38.95%	36.34%	36.44%	36.34%	34.66%	35.71%
2	34.66%	36.13%	38.74%	38.12%	38.64%	35.81%	35.71%	36.02%	36.44%	34.45%
3	35.29%	35.81%	35.92%	36.65%	34.76%	33.93%	35.50%	36.02%	35.81%	35.81%
4	35.08%	36.34%	37.59%	36.65%	37.17%	34.55%	36.96%	36.34%	36.96%	36.86%
5	34.35%	36.34%	35.81%	35.50%	36.34%	34.87%	34.66%	35.18%	34.97%	36.44%
Mean	35.33%	36.40%	37.26%	37.20%	37.17%	35.10%	35.85%	35.98%	35.77%	35.85%

1 second										
	Classification					**Prediction**				
♯ of Split	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$
1	28.69%	30.02%	31.35%	34.43%	34.73%	33.29%	35.55%	36.17%	36.17%	36.78%
2	29.71%	30.64%	31.35%	30.23%	30.23%	31.97%	31.25%	31.76%	31.15%	32.27%
3	31.45%	33.09%	32.17%	33.71%	31.86%	30.84%	29.41%	29.92%	30.94%	29.10%
4	29.51%	30.12%	32.38%	31.56%	31.05%	28.38%	30.12%	31.66%	31.66%	32.17%
5	29.41%	30.74%	30.43%	31.56%	32.89%	30.23%	31.45%	32.68%	32.17%	33.61%
Mean	29.75%	30.92%	31.54%	32.30%	32.15%	30.94%	31.56%	32.44%	32.42%	32.79%

Table 4.13: Classification with KNN on Audio Features considering Δ of 1 or 2 seconds.

2 seconds				
	Classification		**Prediction**	
♯ of Split	Linear Kernel	Rbf Kernel	Linear Kernel	Rbf Kernel
1	34.24%	40.52%	34.35%	40.21%
2	33.19%	40.73%	35.39%	39.90%
3	31.83%	39.48%	34.56%	39.37%
4	33.61%	40.21%	36.34%	36.54%
5	31.83%	39.37%	36.13%	36.75%
Mean	32.94%	40.06%	35.35%	38.55%

1 second				
	Classification		**Prediction**	
♯ of Split	Linear Kernel	Rbf Kernel	Linear Kernel	Rbf Kernel
1	31.76%	35.66%	33.50%	33.61%
2	31.97%	34.22%	33.71%	35.14%
3	31.25%	36.48%	30.53%	33.81%
4	32.99%	36.99%	31.56%	35.25%
5	30.12%	34.12%	33.30%	33.50%
Mean	31.62%	35.94%	32.52%	34.26%

Table 4.14: Classification with SVM on Audio Features considering Δ of 1 or 2 seconds.

computed from a Short Time Fourier Transform, such as MFCC or Spectrograms. In general, spectral features represent audio in a frequency domain and usually obtain better accuracy than temporal features.

MFCCs [17] are among the most used features in speech recognition [162, 163]. These features are based on human hearing perception, and have been used also for the tasks of activity recognition [113, 114] and sound classification [164, 165]. The MFCC algorithm can be summarized as follows. Given an audio signal, the algorithm cut it into short windows containing N samples. The previous and the next windows are overlapped by M samples. To keep continuity of the signal, each frame is multiplied by a Hamming window and then the Fast Fourier Transform (FFT) is applied to compute the magnitude frequency of each window. The Mel-spaced filterbank is computed by using triangular band pass filters in order to get smooth magnitude spectrum. By multiplying each filterbank with the power spectrum, the filterbank energy is computed and the log function is applied to each of them. Finally, the Discrete Cosine Transform (DCT) is computed.

A Spectrogram is a time-frequency representation of audio, that allows to represent the audio signal as a bidimensional one. To compute the spectrogram, the signal is divided into segments of equal length and then the Short Time Fourier Transform is applied to extract the spectrum of each segment. Since spectrograms can be turn on visual representations of audio signals, many papers exploit common neural network architecture, defined for images to these representations. For example, in [87] the log-frequency scale is extracted from the spectrogram and then the U-Net architecture [166] is applied. This convolutional neural network splits the input in K components containing different features of the input sound in order to localize the high resolution features. In [167], it is proposed a variant of VGG model, called VGG-ish, trained only on audio dataset and used to classify audio signal.

We considered the audio modality and built a feature vector made of the concatenation of zero-crossing rate (ZCR) and root mean square energy (RMSE) and tested it with a 5-fold validation. Table 4.10 compares the results for activity classification and anticipation which are obtained with K-NN for different values of K by considering audio representation.

The accuracy values are not up to what we had hoped for. We supposed that audio signal would have helped in recognizing some activities, such as typing or

walking, but results were quite low. This could be due to the following motivations:

- the dataset is unbalanced with respect to the classes despite is balanced with respect transitions.
- the features representation of audio is not suitable for this task.
- the duration of 1 second for audio clips is too short.

All these hypothesis were investigated in the following.

Dataset. As previously discussed, the ST Multimodal Dataset has a balanced past/future transactions (i.e., couples of activities), but it is unbalanced if we consider sequences of single activities (past or future). In Table 4.12, we show the confusion matrix obtained performing the classification on the ZCR and RMSE audio features by using K-NN classifier. All activities are confused with the activity "Desk" which is the most represented class. It is very expensive and hard to balance the data, because balancing the sequences with respect to the past or future activities, the transitions become unbalanced.

Features. Since ZCR and RMSE features did not give good results for classification, we tried to identify the best audio features in a simpler classification scenario discarding anticipation scenario. To this aim, we extracted audio features only on sequences related to the future and considered a subset of the activities ("Typing", "Stairs" and "Walking"). In particular, for training 252 sequences for each activity were selected whereas the test set and the validation set have 84 clips for each activity respectively. The following features were extracted:

- Zero Crossing (ZCR) and Root Mean Square Energy (RMSE);
- MFCC;
- Spectrogram.

Table 4.11 lists the achieved results.

Features are divided into time intervals and are represented in a temporal pyramid fashion. A K-NN classifier with different values of K and an SVM classifier with

linear and RBF kernels have been used as classifiers. MFCC features represented with temporal pyramid (2^{nd} row), give higher values of accuracy than the other representations.

Duration. As mentioned in previous section, we consider a Δ equal to 1 or 2 seconds to produce past/future sequences. Experiments discussed so far are related to sequences with $\Delta = 1$. We hence performed test with sequences of longer duration (i.e., $\Delta = 2$).

In previous test, we observed that MFCC with temporal pyramid gave the best values of accuracy. Hence, we considered only these features to perform classification tests on sequences of 2sec. We used a 5-fold cross-validation procedure with training set composed by 2867 sequences whereas test set and validation set composed by 955 sequences. A comparison of classification accuracy values of audio clips with duration of 1 sec and 2 sec is given in the Table 4.13 and Table 4.14 for KNN and SVM respectively.From the obteined results, it is clear that increasing the duration of audio clips from 1 to 2 seconds, the values of accuracy increase for both classification and anticipation tasks.

4.4 Experimental Results on ST Multimodal Dataset

This section reports experimental settings and results obtained on the ST Multimodal Dataset. Specifically, we aim to assess performances of activity anticipation exploiting representations computed on multimodal signals. To better understand anticipation results, we compare them with respect the task of classification. The difference among anticipation and classification tasks is related to the observed past of the sequence which we want to known the class of activity. For classification task feature can be extracted from the whole sequences, i.e., the sequences is first observed and then classifiers to understand the class of the future activity. For the anticipation task, features are extracted only on the part related to the past of each sequence to infer the future activity (i.e., features on the part of future are not used for inference). We compare the triplet representation on multimodal signals with respect to our previous Section 4.1.1 which we consider a baseline to be improved.

Past	Future	Training Set	Validation Set	Test Set
Desk	Reading	242	81	81
Desk	Typing	243	81	81
Desk	Standing	241	81	81
Reading	Desk	241	81	81
Reading	Typing	248	82	82
Sitting	Desk	240	80	80
Stairs	Walking	250	83	83
Standing	Walking	239	80	80
Typing	Desk	246	82	82
Typing	Reading	242	81	81
Walking	Sitting	238	80	80
Walking	Stairs	252	84	84

Table 4.15: Number of sequences for each transition in training, validation and test set.

Activity	Past	Future
Desk	726	727
Reading	489	484
Sitting	240	238
Stairs	250	252
Standing	239	241
Typing	488	491
Walking	490	489

Table 4.16: Number of sequences for each activity in training set.

4.4.1 Dataset

In our experiments, ST Multimodal Dataset, presented in Chapter 3, was used to evaluate our pipeline. The dataset has been partitioned to use 2922 transactions as training set, and 976 sequences for each set to be used as validation and test. The collected dataset is balanced about the past-future couples, as shown in the Table 4.15, but, if we consider the number of clips for each activity, it is strongly unbalanced, as listed in Table 4.16. For example, the number of sequences of activity labelled "Desk" is 726 in the training set and 243 in the validation and test set, whereas only 250 sequences in the training set and 83 clips in the validation and test set represent the activity "Stairs". Therefore "Desk" is three times greater than the activity "Stairs".

4.4.2 Setup

Auto-encoder is trained for 200 epochs and the batch size is equal to 500 whereas for Triplet Network, 100 epochs and a batch size of 300 samples are considered. In both cases, the Adam optimizer is used with an exponential decreasing of the learning rate starting from 0.001.
K-NN classifier and SVM classifier were used to evaluate proposed representations. Different values of K were used to perform K-NN whereas linear kernel and Rbf kernel were set for SVM.

4.4.3 Rotation of sensors

Data augmentation is an important technique that allows to generate synthetic data useful to train a model and to prevent overfitting, without collecting other data. In our case, this is a fundamental step because collecting human activity takes a long time. So, to improve the accuracy results of our baseline, a data augmentation technique was applied to the inertial data during training.

In the state of the art, there are not many papers that address data augmentation for wearable sensors. In [168], many data augmentation techniques are implemented, such as rotation, which simulates the different sensor placement, permutation, where the data are sliced into segments with the same length and then each of them are permuted to create a new window. Time-warping perturbs the temporal location of data by distorting the time intervals between samples. Scaling method consists in applying a random scalar to change the magnitude of the data, whereas magnitude-warping changes the magnitude of each sample by convolving the data window with a smooth curve varying around one. Finally, jittering is an approach that adds noise to the data.

The aforementioned methods are similar to the techniques used to augment image and video data but an approach specific for wearable sensor is described in [169]. This method considers the physical constraint of the wearable sensor when augmenting data. To add data collected with various rotation angles to training set, they proposed to multiply the sensor data by a 3-dimensional rotation matrix.

In our case, the BlueCoin sensor can be worn in two different ways, as shown in Figure 4.10 a) and b), this causes a rotation only around the z-axis (see Figure 4.10

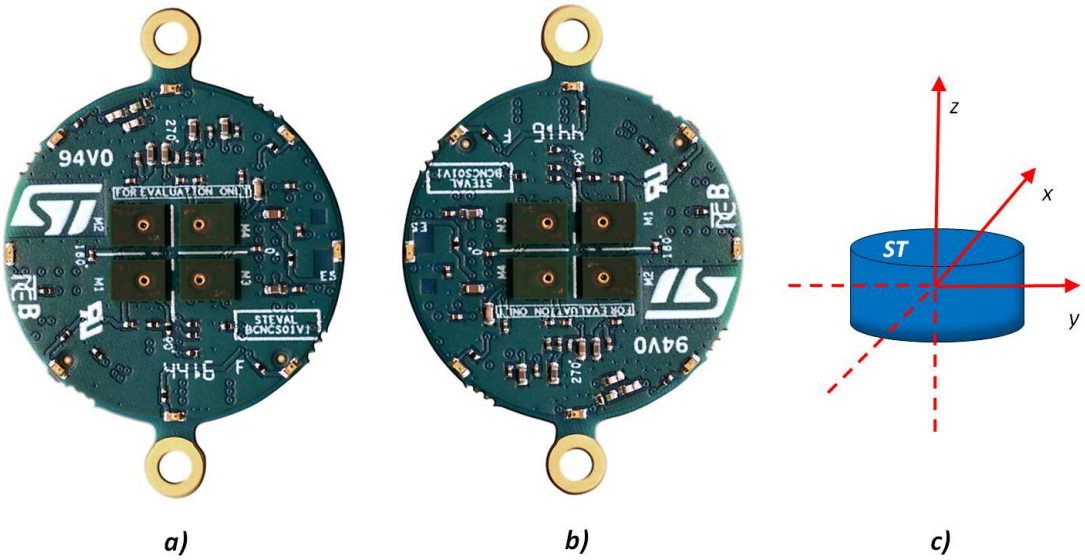


Figure 4.10: Possible ways to wear the Bluecoin sensor (a) and b)) and its axis orientation (c)).

c)). So, inspired by [169], we multiply our 3-dimensional sensor data (accelerometer, gyroscope, magnetic field) by the following rotation matrix:

$$\begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where α is equal to 180°.
The augmented data are concatenated to the training set data and then they are classified and predicted with K-NN and SVM classifier for different values of K and different kernels, respectively. The results are listed in the Tables 4.17 and 4.18, they are also compared with the results of the data without augmentation.

The K-NN results suggest that the data without rotation are classified and predicted better than the data with augmentation. For example, for $K = 3$ and considering the data augmentation, the mean values are 51.72% in classification and 48.24% in anticipation whereas, considering the collected data, the mean results are 52.58% in classification and 48.28% in prediction. With SVM, the mean results of augmented data are higher then accuracy values of collected data but not so much higher to justify the augmentation of the data. Indeed, considering the RBF kernel, the mean results of rotated sensor are 59.02% in classification and 53.30%

Sensors with Rotation										
	Classification					**Anticipation**				
♯ of Split	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$
1	51.33%	51.02%	54.20%	54.00%	54.10%	49.08%	50.31%	49.29%	48.36%	47.85%
2	51.84%	52.46%	54.10%	52.46%	53.07%	46.82%	47.34%	50.10%	50.00%	49.69%
3	52.05%	52.46%	55.12%	54.92%	56.35%	45.49%	47.23%	47.95%	48.67%	47.85%
4	51.84%	50.62%	52.87%	52.15%	52.56%	47.44%	47.75%	48.98%	48.36%	49.08%
5	51.13%	52.05%	51.13%	52.56%	52.97%	49.49%	48.57%	49.69%	51.33%	51.33%
Mean	51.84%	51.72%	53.48%	53.22%	53.81%	47.66%	48.24%	49.20%	49.59%	49.16%

Sensors without Rotation										
	Classification					**Anticipation**				
♯ of Split	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 1$	$K = 3$	$K = 5$	$K = 7$	$K = 9$
1	52.05%	51.64%	55.02%	55.43%	54.61%	49.49%	48.87%	48.26%	47.95%	48.16%
2	52.36%	52.87%	54.30%	52.77%	52.87%	46.41%	47.23%	49.38%	49.69%	51.43%
3	53.38%	52.36%	54.61%	54%	53.69%	44.98%	47.64%	47.95%	48.16%	47.85%
4	53.38%	52.46%	54.1%	52.87%	54.71%	48.05%	48.26%	48.77%	48.67%	50.51%
5	51.02%	53.59%	52.77%	52.05%	53.18%	50.21%	49.39%	49.18%	52.05%	51.13%
Mean	52.44%	52.58%	54.16%	53.42%	53.81%	47.82%	48.28%	48.71%	49.30%	49.82%

Table 4.17: Rotation of sensors: K-NN.

Sensor with Rotation				
	Classification		**Anticipation**	
♯ of Split	Linear Kernel	Rbf Kernel	Linear Kernel	Rbf Kernel
1	47.85%	60.25%	44.26%	55.94%
2	47.03%	57.99%	44.78%	52.15%
3	48.16%	58.71%	43.14%	50.21%
4	47.13%	57.79%	43.75%	54.00%
5	46.52%	60.35%	42.42%	54.20%
Mean	47.34%	59.02%	43.67%	53.30%

Sensor without Rotation				
	Classification		**Anticipation**	
♯ of Split	Linear Kernel	Rbf Kernel	Linear Kernel	Rbf Kernel
1	46.41%	59.43%	43.65%	55.74%
2	44.77%	56.66%	46.62%	51.95%
3	47.64%	58.91%	43.85%	51.13%
4	45.90%	57.07%	43.55%	52.97%
5	46.62%	60.14%	41.80%	54.10%
Mean	46.27%	58.44%	43.89%	53.18%

Table 4.18: Rotation of sensors: SVM.

in prediction, whereas the accuracy values of data without rotation are 58.44% in classification and 53.18% in prediction.

4.4.4 Baseline

As described in the Section 4.2, a comparison between the accuracy values in classification and in prediction defines our baseline. In classification, the features extracted from future clips are classified, whereas in prediction the features related to past sequences and the labels of the future activities are considered.

The accuracy results, obtained with SVM and K-NN classifier, are listed in the Tables 4.19 and 4.20, respectively.

In Table 4.19, it can be observed that the most information comes from video but,

	Classification		**Anticipation**	
Modality	**Linear Kernel**	**RBF**	**Linear Kernel**	**RBF**
Video	66.70%	70.47%	61.35%	66.49%
Audio	31.62%	35.49%	32.52%	34.26%
Sensors	46.27%	58.44%	43.89%	53.18%
Video+Sensors	69.37%	72.89%	68.89%	70.57%
Video+ Audio	67.27%	70.72%	61.52%	67.58%
Audio+Sensors	46.29%	58.83%	43.03%	54.04%
Video+Audio+Sensors	69.55%	73.75%	64.06%	70.29%

Table 4.19: Baseline Results: SVM.

	Classification					**Anticipation**				
Modality	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$	$k=1$	$k=3$	$k=5$	$k=7$	$k=9$
Video	60.02%	59.81%	59.51%	59.53%	59.61%	61.78%	61.31%	60.57%	60.45%	59.73%
Audio	29.75%	30.92%	31.54%	32.30%	32.15%	30.94%	31.56%	32.44%	32.42%	32.79%
Sensors	52.44%	52.58%	54.16%	53.42%	53.81%	47.82%	48.28%	48.71%	49.30%	49.82%
Video+Sensors	62.95%	62.66%	62.40%	61.72%	61.58%	64.18%	63.15%	63.09%	62.32%	61.87%
Video+Audio	60.33%	60%	59.96%	59.76%	59.61%	62.05%	61.72%	60.72%	60.08%	60.29%
Audio+Sensors	54.49%	55.12%	55.57%	55.27%	54.63%	51.29%	52.11%	52.29%	52.19%	51.02%
Video+Audio+Sensors	63.24%	62.79%	62.42%	61.97%	61.76%	64.12%	63.61%	63.14%	62.27%	62.15%

Table 4.20: Baseline Results: K-NN.

combining this modalities with others, the accuracy values increase. Indeed, with Rbf kernel, the video classification accuracy is 70.47% but, for example, considering the combination between video and sensor, the accuracy result is 72.89%. Audio

modality does not seem to be useful for action anticipation task because the accuracy values in classification and in Anticipation are not so higher than the other modalities. With K-NN classifier, the same conclusions can be done.

To our experiences, the accuracy results in classification are higher than those in prediction. With ST Multimodal dataset, it can be observed that this is true by considering SVM classifier, whereas with K-NN classifier the prediction results are higher than those in classification but the gap between them is not so large. For example, in the third split for $K = 3$, the following accuracy results are achieved: 61.17% in classification and 64.14% in prediction. The corresponding confusion matrices are shown in the Tables 4.21 and 4.22. As it can be observed, some activities

	Desk	Reading	Typing	Walking	Sitting	Standing	Stairs
Desk	119	44	48	10	3	19	0
Reading	29	87	44	0	0	2	0
Typing	43	36	83	0	0	1	0
Walking	8	3	0	125	8	7	12
Sitting	7	0	0	7	59	7	0
Standing	11	1	0	14	9	46	0
Stairs	0	0	0	6	0	0	78

Table 4.21: Transition Matrix: classification of future sequences.

	Desk	Reading	Typing	Walking	Sitting	Standing	Stairs
Desk	144	28	43	13	10	5	0
Reading	28	84	50	0	0	0	0
Typing	32	39	91	0	0	1	0
Walking	11	1	2	125	13	4	7
Sitting	5	1	0	6	67	0	1
Standing	13	4	2	18	4	40	0
Stairs	0	0	0	7	2	0	75

Table 4.22: Transition Matrix: classification of past clips (Anticipation).

are obvious, such as, Sitting, Standing and Stairs, indeed they are correctly classified and predicted. In prediction (Table 4.22), the significant improvement is given by the "Desk" class that is successfully predicted 144 times whereas it is properly classified 119 times (Table 4.21). To better understand the Table 4.22, a study of how many sequences are correctly predicted is shown in the Table 4.23. Since "Desk"

is properly anticipated 144 times and it has three possible past sequences, we may ask ourselves which is the corresponding one. So, K-NN predicts it 52 times from Typing, 39 times from Reading and 53 times from Sitting.

Past	Fut	Correct	Mistake
Desk	Reading	30	51
Reading	Typing	50	32
Typing	Desk	52	30
Desk	Typing	41	40
Typing	Reading	54	27
Reading	Desk	39	42
Walking	Stairs	75	9
Stairs	Walking	75	8
Walking	Sitting	67	13
Sitting	Desk	53	27
Desk	Standing	40	41
Standing	Walking	50	30

Table 4.23: Study of how many sequences are correctly predicted.

4.4.5 Auto-encoder and Triplet Network

As sad in the Section 4.2, to anticipate the future activity by observing only data related to past sequences, it is important to fill the gap between the accuracy values in classification and in prediction. To this aim, two different architectures are considered: Auto-encoder and Triplet Network. In both cases, three 1D CNNs are used. For the first network, the number of filters is 32,64 and 4 respectively and a max-pooling layer is used to reduce the size of the output of each convolutional layer. The output of encoder stream has dimension of 4040. For Triplet Network, 32,64 and 2 are the used filters and, to prevent overfitting, a kernel regularize equals to 0.001 and a dropout of 0.2 are applied to each convolution layer during training procedure. The size of the output of a branch of this network is 2020.

Auto-encoder. As it is possible to note in Table 4.16, "Desk" is a most represented class in the dataset. So, since the gaol is to anticipate the future action, to train an Auto-encoder, the training set is balanced by considering the future activities. In particular, all sequences are added in order to have the same number

of samples of Desk activity (727), therefore the following numbers of sequences for each activity are added:

- Desk: 0;
- Reading: 243;
- Typing: 236;
- Walking: 238;
- Sitting: 489;
- Standing: 486;
- Stairs: 475.

In total, 2167 sequences are added and the training set has 5089 samples.

Tables 4.24 and 4.25 show the achieved results on action anticipation with SVM and K-NN classifier, respectively. With K-NN classifier, the accuracy results are

Classification		**Anticipation**			
Baseline		**Baseline**		**Auto-encoder**	
Linear Kernel	**RBF**	**Linear Kernel**	**RBF**	**Linear Kernel**	**RBF**
69.55%	73.75%	64.06%	70.29%	62.11%	68.18%

Table 4.24: Auto-encoder: SVM Results.

	Classification	**Anticipation**	
K	**Baseline**	**Baseline**	**Triplet**
1	63.24%	64.12%	64.03%
3	62.79%	63.61%	63.67%
5	62.79%	63.14%	62.52%
7	61.97%	62.27%	61.97%
9	61.76%	62.15%	61.19%

Table 4.25: Auto-encoder: K-NN Results.

between the baseline accuracy values in classification and in prediction. For example, for K=1, we have 63.24% in classification and 63.61% in prediction, with auto-encoder 64.03% is achieved. For K=5 and K=9, the results are lower than

those in baseline whereas, for K=7, we have the same baseline values in classification. The accuracy values obtained with SVM classifier do not reach the baseline results.

Triplet Network. To build a better embedding space where past and future sequences, semantically correlated, are close, we exploited a Triplet network. Euclidean metric is used to compute the distance between the transformation of the inputs to minimize the following loss function:

$$\mathcal{L}(y) = \max(D(x, x^+) - D(x, x^-) + margin, 0) \tag{4.3}$$

where D is the euclidean metrics between two feature representations and $margin$ is the margin to respect between positive and negative pairs.

To train the Triplet Network, it was necessary to create positive and negative couples with binary labels (i.e., possible transaction vs not possible transaction) by combining the sequences of the past and the future. Indeed, each past sequence is paired with all the possible future activities, excluding those which past and future are equal. Therefore the couples like Desk/Desk, Reading/Reading, and so on, are not generated. If the past and future couple belong to one of the possible transitions defined in Chapter 3 and listed in the first two columns of Table 4.23 (e.g. the past activity is Desk and the following action is Typing), we assign a label 1, otherwise if the past and future sequences are not possible (e.g. the past activity is Desk and the future is Walking), we assign a label 0 for training purposes.

To create a balanced dataset, for each past sequences 30000 true couples and an equal number of false pairs have been generated. For example, for Desk class, 10000 samples have been created for each of the following transitions: Desk/Reading, Desk/Typing, Desk/Sitting, as possible transactions (label 1), and Desk/Walking, Desk/Standing and Desk/Stairs as not possible transaction (label 0). Since Stairs has only Walking as future activity, 30000 samples are considered only for the possible transition with label 1, whereas 6000 samples are created for the other 5 false transitions (Stairs/Desk, Stairs/Reading, Stairs/Typing, Stairs/Sitting, Stairs/Standing). In this way, 210000 couples with label 1 and 210000 pairs with label 0 are considered.

Tables 4.26 and 4.27 report the achieved results with SVM and K-NN classifiers,

Classification		**Anticipation**			
Baseline		**Baseline**		**Triplet**	
Linear Kernel	**RBF**	**Linear Kernel**	**RBF**	**Linear Kernel**	**RBF**
69.55%	73.75%	64.06%	70.29%	57.52%	63.32%

Table 4.26: Triplet Network: SVM Results.

	Classification	**Anticipation**	
K	**Baseline**	**Baseline**	**Triplet**
1	63.24%	64.12%	64.65%
3	62.79%	63.61%	64.73%
5	62.79%	63.14%	64.14%
7	61.97%	62.27%	64.55%
9	61.76%	62.15%	64.18%

Table 4.27: Triplet Network: K-NN Results.

respectively. With K-NN, all baseline values for classification and anticipation are improved. The best improvement is given by KNN with $K = 7$ and $K = 9$ where the obtained results exceed 2% the baseline values. SVM classifier does not reach the baseline values, indeed considering RBF Kernel, the values for classification is 73.75% and the one for anticipation is 70.29% with baseline representation, whereas, the accuracy results with a Triplet Network is 63.32%. This may be due to the fact that the hyperplanes generated with SVM fail to separate the features projected in the embedding space, built with the Triplet Network which is specialised to work with Euclidean space used by KNN.

The achieved results suggest that the selected activities can be anticipated by considering both different modalities and a large number of transitions from past to future.

4.5 Summary

This chapter presents results on action anticipation from multimodal data. Our pipeline is evaluated on Stanford-ECM Dataset and on ST Multimodal Dataset. We compared the performances of different architectures and classifiers.

Two different situations are evaluated. With Stanford-ECM Dataset, the anticipation from a generic unknown activity to a specific action is considered whereas,

with ST Multimodal Dataset, a future action is anticipated from a specific activity.

Preliminary results on Stanford-ECM Dataset suggest that multi-modality improves both classification and prediction tasks, but we could not deeply take advantage of deep learning approaches on multi-modal data due to a very limited dataset for training the methods.

Populating a large dataset and improving the pipeline seem to suggest that anticipate the future action is possible. Indeed, the number of transition of ST Multimodal Dataset is almost sixteen times higher than Stanford-ECM Dataset. Moreover, Triplet Network is able to fill the gap between the accuracy values in classification and in prediction and to overcame them.

Chapter 5

Conclusions

The main contribution of this thesis is related to the investigation of approaches of Multimodal Learning and Action Anticipation. Anticipating the near future and combining the information coming from different senses, such as eyes, skin etc., are natural tasks for humans but they represent an open issue in Computer Vision and in Deep Learning. Given the difficulties, action anticipation and multimodal learning task have always been treated separately, as shown in Chapter 2.

The definition of the problem is really challenging, since it is associated to a creation of a multimodal pipeline which combines the input features in order to anticipate the future action. Since starting from past experiences, our brain is able to associate similar semantic events and/or situation in order to generalize and understand the next action to be performed, we tried to anticipate future activity by training a neural network with past sequences that represent human past experiences. So, studying the data sequences, it is possible to detect a point, called transition point, where the activity changes. This allowed us to adapt the dataset and define the problem as follows. Given the time related to the action change, activities were cut around its in order to generate past and future sequences. The past clips were used to train a Neural Network to predict the future label by observing data before the activity starts.

Section 2.5 revealed that many multimodal datasets available in the state of the art were created for action recognition task and there is not any multimodal dataset for action anticipation. Moreover, the definition of our problem involved the use of more transitions from a past to a future activity but the datasets presented in literature do not have enough to train a neural network. So, Chapter 3 described

the collection procedure of a multimodal dataset that we created for action anticipation task, called ST Multimodal Dataset, which was collected with a smartphone camera and a BlueCoin board. It comprises the following modalities: video, audio, tri-axial accelerometers, tri-axial gyroscope, tri-axial magnetic field, pressure and temperature. 7 daily activities that are quite common in an office are acquired from 19 participants.

The approaches presented in Section 4.1 and in Section 4.3 is able to combine visual data and sensor information and project them in a new feature space (embedding space) where the semantically correlations between data are considered. In other word, past and future sequences semantically correlated are represented close to each other in an embedding space, otherwise they are projected in two different places of the space. Our approaches exploited different deep architectures (Autoencoder and Siamese Network in 4.1, Autoencoder and Triplet Network in Section 4.3) which were evaluated with SVM and K-NN classifiers for different kernels and for different values of K, respectively. Moreover, to identify the contribution of each signal, as baseline, each modality and combination of all of them were classified. Moreover, the comparison between the accuracy values in classification and in anticipation was considered. In classification, the features related to future clips were classified whereas the past features and the future labels were considered in anticipation.

The achieved results suggest the following insights. The first results have been shown that, comparing inputs with each other, the higher accuracy value is given by visual modality. This value is consistent with human senses. Indeed, humans rely on sight to capture main significant information. Combining the different modalities between them, the accuracy values increases. In general, it was also observed that the results in classification were higher than those in anticipation. This allowed us to define as goal the filling of the gap between baseline accuracies by using a Neural Network. In this case, results have been shown that, populating a large dataset and improving the pipeline, the anticipation of the future action is possible. For example, with a Triplet Network and a KNN classifier, the achieved results overcome the baseline results by 2%. Moreover, the test results suggest that using video, sensor data and their combination can achieve higher accuracy in classification and in anticipation respect to the audio signals.

Future works could be devoted to extend the dataset collecting more sequences and to acquire bigger labelled multimodal datasets considering different environments and activities. In this thesis, early concatenation technique was used to concatenate the features of the different data. Different approaches to fuse data collected from different devices can be taken into account. For example, it could be used an attention mechanism in order to weight each modality. Furthermore, other neural network architectures, such as RNNs and LSTMs that keep, in the hidden units, information related to the previous elements of the sequences, could be considered to address the problem of action anticipation task.

The achieved results are very interesting because they allow to consider new applications. In fact, anticipating future actions from different data may improve fields such us health care and surveillance. For example, in a driving system is possible to capture not only visual data from a camera installed in a car, but this device can collect other signals, such as GPS, inertial sensor, audio from microphones, and so on. They could be used to improve the existing system to predict accidents even before they occur. In addition, research can be extended to applications aimed at increasing the interaction between humans and robots or simply between robots.

Appendix A

A Digital Countryside Notebook for Smart Agriculture and Oranges Classification

In this paper, we propose a system useful to monitor land property based on IoT cloud platform. It involves the dislocation of a series of sensors on the agricultural field in order to detect all the interesting parameters to monitor and manage the land property, such as temperature, humidity, solar irradiation, etc. Through a wireless network, a cloud platform receives and examines these data. The proposed system represents a first prototype with respect to a classic countryside notebook. The farmers usually use this document to report every operation on the product, ensuring its traceability. More specifically, we propose a digitalization of countryside notebook and develop a "K-NN Oranges Variety Classifier" to help the land owner for the automatic labelling of the acquired images of an orange plant among different varieties. The mobile App, shown in Figure A.1, was developed to work on an Android operating system.

Figure A.1: Screenshots of our countryside notebook.

Appendix B

Generalised Gradient Vector Flow for Content-aware Image Resizing

Image retargeting is devoted to preserve the visual content of images with a proper resizing, removing vertical and/or horizontal paths of pixels which contain low semantic information. In this paper, we present a new method for image retargeting which is based on GGVF. We assess and investigate the importance of one of the main involved parameter (K) of GGVF, which balances the smoothing term and data term. The proposed approach has been compared with respect to a method based on GVF [170] and a seam carving approach [171] and it has been tested by considering a data set of 1000 images and varying the percentage of resizing from 10% to 50% and for different values of the aim involved parameter K.

Results show that our algorithm better preserves the important information compared to GVF and Seam Carving approaches, as shown in Figure B.2 and Figure B.3. The three algorithms have different behaviours. In particular, comparing the seems generated by the proposed algorithm (3^{th} column) and the ones generated by the GVF scheme (5^{th} column) or by the seam carving approach (7^{th} column), is possible to observe that the methods of the state of the art remove information from the object introducing deformations and distortions on the image, whereas the GGVF approach preserves the visual content of the scene by maintaining both size of the objects and the details related the visual stimuli of textures and edges.

To evaluate the performance of our algorithm for different values of K, for each $i-th$ image, we considered the number of pixels in its binary mask p_i^{bm} and the number of successfully preserved pixels after the application of the Seam Carving (SC), the GVF and the GGVF methods, denoted as n_i^{SC}, n_i^{GVF} and n_i^{GGVF} respectively. The

quality of a resized image is evaluated by considering the ratio between n_i^m and p_i^{bm}:

$$q_i^m = \frac{n_i^m}{p_i^{bm}} \tag{B.1}$$

where $m \in \{SC, GVF, GGVF\}$ is the resizing method applied to the input image. Based on these definitions, the following evaluation score is computed:

$$Score_2 = \frac{1}{N} \sum_{i=1}^{|T|} q_i^m \tag{B.2}$$

Table B.1 shows the achieved experimental results in terms of average $Score_2$, by varying the resizing factor and the value of K. The achieved results suggest that there is a relationship between K and the percentage of resizing. However, when the resizing factor is set to extreme values, the performances start to decrease after a certain value of K.

	0.001	**0.05**	**0.75**	**1**	**1.25**
10%	72.7%	74.7%	78.2%	77.7%	77.9%
20%	65.!%	65.6%	67.7%	68.2%	67.6%
30%	56.4%	56.8%	57.6%	58.2%	57.1%
40%	50.3%	51.4%	48.4%	47.7%	46.7%
50%	44.1%	43.9%	38.9%	37.7%	37.7%

Figure B.1: $Score_2$ obtained with Equation B.2.

Figure B.4 shows three examples with a scale factor of 40%. The 2^{th} and 4^{th} columns show the results obtained by GGVF (with the best choice for K), by GVF and by Seam Carving respectively. The values reported under each image are the cost obtained with Equation B.1. The 5^{th} column highlights how our approach better preserves the main object of the input image with respect to other algorithms (6^{th} and 7^{th} column).

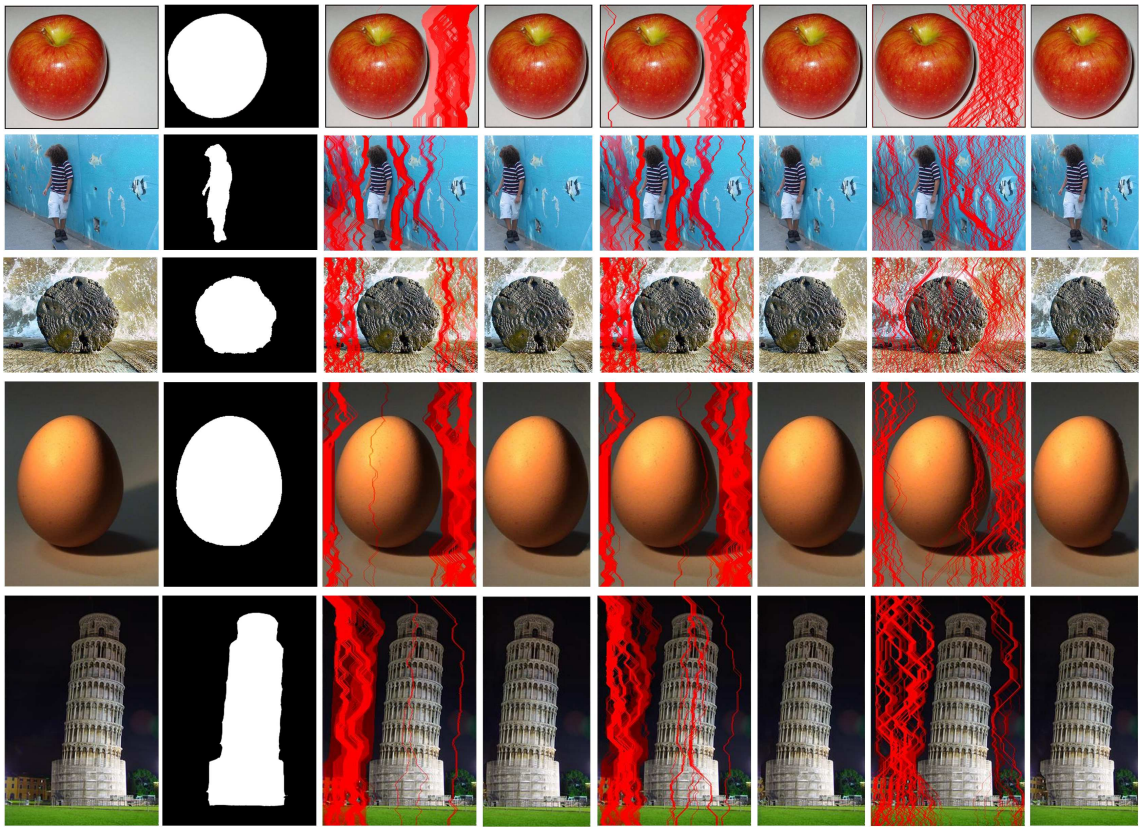


Figure B.2: Examples of image resizing at 70% of the original width. Original image (1^{th} column), binary mask (2^{th} column), seams generated by our approach with $K = 1$ (3^{th} column), our result (4^{th} column), seams generated by GVF (5^{th} column), GVF result (6^{th} column), seams generated by Seam Carving (7^{th} column) and its result (8^{th} column).

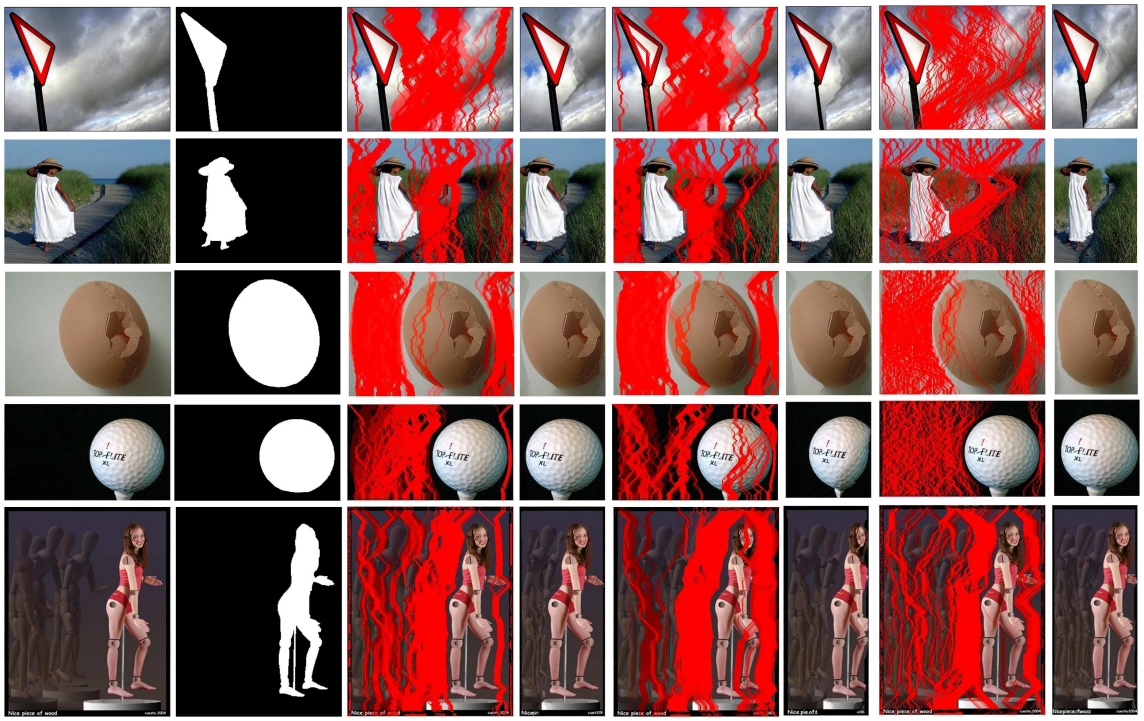


Figure B.3: Examples of image resizing at 50% of the original width. Original image (1^{th} column), binary mask (2^{th} column), seams generated by our approach with $K = 0.05$ (3^{th} column), our result (4^{th} column), seams generated by GVF (5^{th} column) and GVF result (6^{th} column), seams generated by Seam Carving (7^{th} column) and its result (8^{th} column).

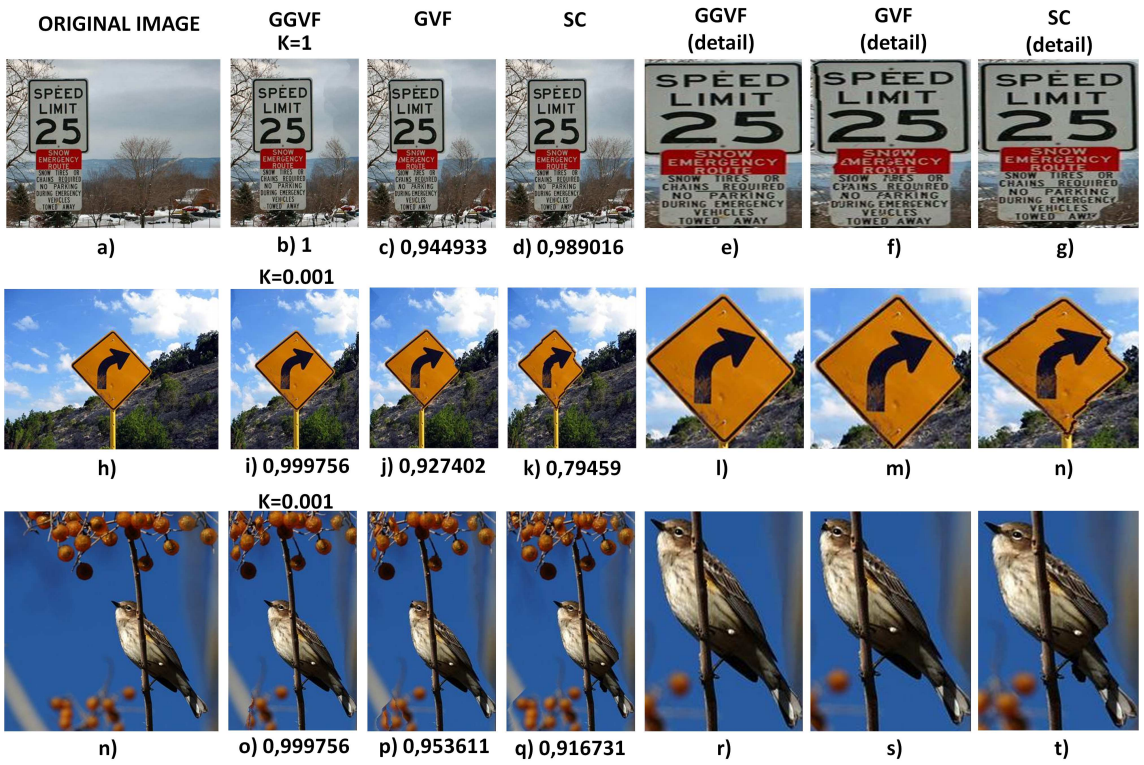


Figure B.4: Examples of image resizing at 40% of the original width. The original images are shown in the first column. The second column reports the resizing results obtained by applying GGVF and the related cost (i.e., Equation B.1). The third column shows the results obtained by GVF, whereas the fourth column reports the Seam Carving results. The last three columns show some details of the outputs obtained by GGVF, GVF and Seam Carving.

Bibliography

[1] H. McGurk and J. MacDonald. "Hearing lips and seeing voices". In: *Nature* 264 (Dec. 1976), pp. 746–748.

[2] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. "Tensor Fusion Network for Multimodal Sentiment Analysis". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1103–1114. DOI: 10.18653/v1/D17-1115. URL: https://www.aclweb.org/anthology/D17-1115.

[3] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. "Multimodal Deep Learning". In: *Proceedings of the 28th International Conference on Machine Learning*. Omnipress, 2011, pp. 689–696. ISBN: 978-1-4503-0619-5. URL: http://dl.acm.org/citation.cfm?id=3104482.3104569.

[4] N. Srivastava and R. Salakhutdinov. "Multimodal Learning with Deep Boltzmann Machines". In: *Journal of Machine Learning Research* 15 (2014), pp. 2949–2980. URL: http://jmlr.org/papers/v15/srivastava14b.html.

[5] Y. Aytar, C. Vondrick, and A. Torralba. "See, Hear, and Read: Deep Aligned Representations". In: abs/1706.00932 (2017). arXiv: 1706.00932. URL: http://arxiv.org/abs/1706.00932.

[6] H. Hotelling. "Relations between two sets of variates ". In: *Biometrika* 28.3-4 (Dec. 1936), pp. 321–377. ISSN: 0006-3444. DOI: 10.1093/biomet/28.3-4.321. eprint: http://oup.prod.sis.lan/biomet/article-pdf/28/3-4/321/586830/28-3-4-321.pdf. URL: https://doi.org/10.1093/biomet/28.3-4.321.

[7] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. "Deep Canonical Correlation Analysis". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA,

USA: JMLR.org, 2013, pp. III–1247–III–1255. URL: http://dl.acm.org/citation.cfm?id=3042817.3043076.

[8] P. Lai and C Fyfe. "Kernel and nonlinear canonical correlation analysis". In: *Int. Journal of Neural Systems* 10 (Nov. 2000), pp. 365 –377. DOI: 10.1016/S0129-0657(00)00034-X.

[9] M. Dalla Mura, S. Prasad, F. Pacifici, P. Gamba, J. Chanussot, and J. A. Benediktsson. "Challenges and Opportunities of Multimodality and Data Fusion in Remote Sensing". In: *Proceedings of the IEEE* 103.9 (2015), pp. 1585–1601. ISSN: 0018-9219. DOI: 10.1109/JPROC.2015.2462751.

[10] K. Noda, H. Arie, Y. Suga, and T. Ogata. "Multimodal integration learning of robot behavior using deep neural networks". In: *Robotics and Autonomous Systems* 62.6 (2014), pp. 721 –736. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2014.03.003. URL: http://www.sciencedirect.com/science/article/pii/S0921889014000396.

[11] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard. "Multimodal deep learning for robust RGB-D object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 681–687. DOI: 10.1109/IROS.2015.7353446.

[12] A. Wali and A. M. Alimi. "Multimodal Approach for Video Surveillance Indexing and Retrieval". In: *Journal of Intelligent Computing* 1 (4 2010), pp. 165–175. arXiv: 1308.1150. URL: http://arxiv.org/abs/1308.1150.

[13] Y. Tang, D. Wu, Z. Jin, W. Zou, and X. Li. "Multi-modal metric learning for vehicle re-identification in traffic surveillance environment". In: *2017 IEEE International Conference on Image Processing (ICIP)*. Sept. 2017, pp. 2254–2258. DOI: 10.1109/ICIP.2017.8296683.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1–9.

[16] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556.

[17] H. Hermansky and N. Malayath. "Spectral basis functions from discriminant analysis." In: *International Conference on Spoken Language Processing*. Jan. 1998.

[18] A. Furnari, S. Battiato, and G. M. Farinella. "Leveraging Uncertainty to Rethink Loss Functions and Evaluation Measures for Egocentric Action Anticipation". In: *European Conference on Computer Vision – ECCV 2018 Workshops*. Ed. by L. Leal-Taixé and S. Roth. Cham: Springer International Publishing, 2019, pp. 389–405. ISBN: 978-3-030-11021-5.

[19] T. Lan, T.-C. Chen, and S. Savarese. "A Hierarchical Representation for Future Action Prediction". In: *European Conference on Computer Vision – ECCV*. Cham: Springer International Publishing, 2014, pp. 689–704. ISBN: 978-3-319-10578-9.

[20] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. "Anticipating Accidents in Dashcam Videos". In: *Asian Conference on Computer Vision*. Ed. by S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato. Cham: Springer International Publishing, 2017, pp. 136–153. ISBN: 978-3-319-54190-7.

[21] H. S. Koppula and A. Saxena. "Anticipating Human Activities Using Object Affordances for Reactive Robotic Response". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 38.1 (Jan. 2016), pp. 14–29. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2430335. URL: http://dx.doi.org/10.1109/TPAMI.2015.2430335.

[22] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella. "Next-active-object prediction from egocentric videos". In: *Journal of Visual Communication and Image Representation* 49 (2017), pp. 401 –411. ISSN: 1047-3203.

DOI: https://doi.org/10.1016/j.jvcir.2017.10.004. URL: http://www.sciencedirect.com/science/article/pii/S1047320317301967.

[23] H. S. Koppula, A. Jain, and A. Saxena. "cipatory Planning for Human-Robot Teams". In: *Experimental Robotics: The 14th International Symposium on Experimental Robotics*. 2016.

[24] J. Mainprice and D. Berenson. "Human-robot collaborative manipulation planning using early prediction of human motion". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 299–306. DOI: 10.1109/IROS.2013.6696368.

[25] N. Duarte, J. Tasevski, M. I. Coco, M. Rakovic, and J. Santos-Victor. "Action Anticipation: Reading the Intentions of Humans and Robots". In: *IEEE Robotics and Automation Letters* abs/1802.02788 (2018). arXiv: 1802.02788. URL: http://arxiv.org/abs/1802.02788.

[26] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *European Conference on Computer Vision (ECCV)* (2018).

[27] C. Vondrick and A. Torralba. "Generating the Future with Adversarial Transformers". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2992–3000. DOI: 10.1109/CVPR.2017.319.

[28] R. Salakhutdinov and G. E. Hinton. "Deep Boltzmann Machines". In: *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida*. 2009, pp. 448–455.

[29] Y. LeCun, Y. Bengio, and G. E. Hinton. "Deel Learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[30] K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei. "Jointly Learning Energy Expenditures and Activities Using Egocentric Multimodal Signals". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6817–6826. DOI: 10.1109/CVPR.2017.721.

[31] G. E. Hinton and R. R. Salakhutdinov. "Reducing the dimensionality of data with neural networks". In: *Science* 313.5786 (July 2006), pp. 504–507. DOI: 10.1126/science.1127647.

[32] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. "Extracting and Composing Robust Features with Denoising Autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: ACM, 2008, pp. 1096–1103. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390294. URL: http://doi.acm.org/10.1145/1390156.1390294.

[33] X. Peng, Y. Li, X. Wei, J. Luo, and Y. L. Murphey. "Traffic sign recognition with transfer learning". In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017, pp. 1–7. DOI: 10.1109/SSCI.2017.8285332.

[34] X. lu, Y. Tsao, S. Matsuda, and C. Hori. "Speech enhancement based on deep denoising Auto-Encoder". In: *Proc. Interspeech* (Jan. 2013), pp. 436–440.

[35] D. E. Rumelhart and J. L. McClelland. "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. MITP, 1987. URL: https://ieeexplore.ieee.org/document/6302929.

[36] M. Ranzato, Y.-L Boureau, and Y LeCun. "Sparse feature learning for deep belief networks". In: *Advances in Neural Information Processing Systems* 20 (Jan. 2008), pp. 1185–1192.

[37] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *J. Mach. Learn. Res.* 11 (2010), pp. 3371–3408. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1756006.1953039.

[38] R. Salakhutdinov and G. Hinton. "Semantic hashing". In: *International Journal of Approximate Reasoning* 50.7 (2009). Special Section on Graphical Models and Information Retrieval, pp. 969 –978. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2008.11.006. URL: http://www.sciencedirect.com/science/article/pii/S0888613X08001813.

[39] C. Silberer and M. Lapata. "Learning Grounded Meaning Representations with Autoencoders". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 721–732. DOI: 10.3115/v1/P14-1068. URL: https://www.aclweb.org/anthology/P14-1068.

[40] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. "Signature Verification Using a "Siamese" Time Delay Neural Network". In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. Morgan Kaufmann Publishers Inc., 1993, pp. 737–744. URL: http://dl.acm.org/citation.cfm?id=2987189.2987282.

[41] G. Koch, R. Zemel, and R. Salakhutdinov. "Siamese Neural Networks for One-shot Image Recognition". In: *ICML Deep Learning Workshop*. 2015.

[42] R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality Reduction by Learning an Invariant Mapping". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 2006, pp. 1735–1742. DOI: 10.1109/CVPR.2006.100.

[43] E. Hoffer and N. Ailon. "Deep metric learning using Triplet network." In: *ICLR (Workshop)*. Ed. by Y. Bengio and Y. LeCun. 2015. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2015w.html.HofferA14.

[44] E. Triantafillou, R. Zemel, and R. Urtasun. "Few-Shot Learning Through an Information Retrieval Lens". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 2255–2265. URL: http://papers.nips.cc/paper/6820-few-shot-learning-through-an-information-retrieval-lens.pdf.

[45] R. Eloff, H. A. Engelbrecht, and H. Kamper. "Multimodal One-Shot Learning of Speech and Images". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* abs/1811.03875 (2019). arXiv: 1811.03875. URL: http://arxiv.org/abs/1811.03875.

[46] N. N. Vo and J. Hays. "Localizing and Orienting Street Views Using Overhead Imagery". In: *European Conference on Computer Vision (ECCV) 2016* abs/1608.00161 (2016). arXiv: 1608.00161. URL: http://arxiv.org/abs/1608.00161.

[47] F. Huang, X. Zhang, Z. Li, T. Mei, Y. He, and Z. Zhao. "Learning Social Image Embedding with Deep Multimodal Attention Networks". In: *Thematic Workshops of the 25th ACM Multimedia* abs/1710.06582 (2017). arXiv: 1710.06582. URL: http://arxiv.org/abs/1710.06582.

[48] E. Simo Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. "Discriminative learning of deep convolutional feature point descriptors". English. In: *Proceedings - 2015 IEEE International Conference on Computer Vision, ICCV 2015*. Vol. 11-18-December-2015. United States: Institute of Electrical and Electronics Engineers Inc., Feb. 2016, pp. 118–126. DOI: 10.1109/ICCV.2015.22.

[49] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. "Fully-Convolutional Siamese Networks for Object Tracking". In: *European Conference on Computer Vision (ECCV)* abs/1606.09549 (2016). arXiv: 1606.09549. URL: http://arxiv.org/abs/1606.09549.

[50] R. Tao, E. Gavves, and A. W. M. Smeulders. "Siamese Instance Search for Tracking". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1420–1429. DOI: 10.1109/CVPR.2016.158.

[51] X. Dong and J. Shen. "Triplet Loss in Siamese Network for Object Tracking". In: *Computer Vision – ECCV 2018*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Cham: Springer International Publishing, 2018, pp. 472–488. ISBN: 978-3-030-01261-8.

[52] D. P. Vassileios Balntas Edgar Riba and K. Mikolajczyk. "Learning local feature descriptors with triplets and shallow convolutional neural networks". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by E. R. H. Richard C. Wilson and W. A. P. Smith. BMVA Press, 2016, pp. 119.1–119.11. ISBN: 1-901725-59-6. DOI: 10.5244/C.30.119. URL: https://dx.doi.org/10.5244/C.30.119.

[53] S. Zhang, Q. Zhang, X. Wei, Y. Zhang, and Y. Xia. "Person Re-identification with Triplet Focal Loss". In: *IEEE Access* PP (Dec. 2018), pp. 1–1. DOI: 10.1109/ACCESS.2018.2884743.

[54] E. Ahmed, M. Jones, and T. K. Marks. "An Improved Deep Learning Architecture for Person Re-Identification". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[55] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. "Person Re-identification by Multi-Channel Parts-Based CNN with Improved Triplet Loss Function". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1335–1344. DOI: 10.1109/CVPR.2016.149.

[56] R. R. Varior, M. Haloi, and G. Wang. "Gated Siamese Convolutional Neural Network Architecture for Human Re-identification". In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*. 2016, pp. 791–808. DOI: 10.1007/978-3-319-46484-8_48. URL: https://doi.org/10.1007/978-3-319-46484-8_48.

[57] Y. Sun, X. Wang, and X. Tang. "Deep Learning Face Representation from Predicting 10,000 Classes". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1891–1898. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.244. URL: https://doi.org/10.1109/CVPR.2014.244.

[58] S. Chopra, R. Hadsell, and Y. LeCun. "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 539–546. ISBN: 0-7695-2372-2. DOI: 10.1109/CVPR.2005.202. URL: http://dx.doi.org/10.1109/CVPR.2005.202.

[59] J. Hu, J. Lu, and Y. Tan. "Discriminative Deep Metric Learning for Face Verification in the Wild". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1875–1882. DOI: 10.1109/CVPR.2014.242.

[60] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *CVPR*. 2015, pp. 815–823. URL: https://doi.org/10.1109/CVPR.2015.7298682.

[61] C.-w. Hsu, C.-c. Chang, and C.-J. Linl. "A Partical Guide to Support Vector Classification." In: (2003).

[62] C. M. Bishop. *Patter Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[63] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. "KNN Model-Based Approach in Classification." In: *Lect Notes Computer Sci* 2888 (2003), pp. 986–996.

[64] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz. "Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition". In: *IEEE Sensors Journal* 14.6 (2014), pp. 1898–1903. ISSN: 1530-437X. DOI: 10.1109/JSEN.2014.2306094.

[65] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G. Z. Yang. "Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems". In: *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*. Ed. by S. Leonhardt, T. Falck, and P. Mähönen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 208–212. ISBN: 978-3-540-70994-7.

[66] E. Morvant, A. Habrard, and S. Ayache. "Majority Vote of Diverse Classifiers for Late Fusion". In: *Structural, Syntactic, and Statistical Pattern Recognition*. Ed. by P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 153–162. ISBN: 978-3-662-44415-3.

[67] G. A. Ramirez, T. Baltrušaitis, and L.-P. Morency. "Modeling Latent Discriminative Dynamic of Multi-dimensional Affective Signals". In: *Affective Computing and Intelligent Interaction*. Ed. by S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 396–406. ISBN: 978-3-642-24571-8.

[68] M. Glodek, S. Tschechne, G. Layher, M. Schels, T. Brosch, S. Scherer, M. Kächele, M. Schmidt, H. Neumann, G. Palm, and F. Schwenker. "Multiple Classifier Systems for the Classification of Audio-Visual Emotional States". In: *Affective Computing and Intelligent Interaction*. Ed. by S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 359–368. ISBN: 978-3-642-24571-8.

[69] D. a. J. Shutova Ekaterina andKiela. "Black Holes and White Rabbits: Metaphor Identification with Visual Features". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 160–170. DOI: 10.18653/v1/N16-1020. URL: https://www.aclweb.org/anthology/N16-1020.

[70] Y. Li, M. Liu, and W. Sheng. "Indoor human tracking and state estimation by fusing environmental sensors and wearable sensors". In: *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2015, pp. 1468–1473. DOI: 10.1109/CYBER.2015.7288161.

[71] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. "Multimodal fusion for multimedia analysis: a survey". In: *Multimedia Systems* 16.6 (2010), pp. 345–379. ISSN: 1432-1882. DOI: 10.1007/s00530-010-0182-0. URL: https://doi.org/10.1007/s00530-010-0182-0.

[72] Z. Wu, L. Cai, and H. Meng. "Multi-level Fusion of Audio and Visual Features for Speaker Identification". In: *Advances in Biometrics*. Ed. by D. Zhang and A. K. Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 493–499. ISBN: 978-3-540-31621-3.

[73] H. Gunes and M. Piccardi. "Affect recognition from face and body: early fusion vs. late fusion". In: *2005 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 4. 2005, 3437–3443 Vol. 4. DOI: 10.1109/ICSMC.2005.1571679.

[74] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. "Early Versus Late Fusion in Semantic Video Analysis". In: *Proceedings of the 13th Annual ACM International Conference on Multimedia*. MULTIMEDIA '05. ACM, 2005,

pp. 399–402. ISBN: 1-59593-044-2. DOI: 10.1145/1101149.1101236. URL: http://doi.acm.org/10.1145/1101149.1101236.

[75] K. Liu, Y. Li, N. Xu, and P. Natarajan. *Learn to Combine Modalities in Multimodal Deep Learning*. 2018.

[76] J. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie. "MFAS: Multimodal Fusion Architecture Search". In: *CoRR* abs/1903.06496 (2019). arXiv: 1903.06496. URL: http://arxiv.org/abs/1903.06496.

[77] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Bagher Zadeh, and L.-P. Morency. "Efficient Low-rank Multimodal Fusion With Modality-Specific Factors". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2247–2256. DOI: 10.18653/v1/P18-1209. URL: https://www.aclweb.org/anthology/P18-1209.

[78] K. Cho, A. Courville, and Y. Bengio. "Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks". In: *IEEE Transactions on Multimedia* 17 (July 2015). DOI: 10.1109/TMM.2015.2477044.

[79] A. Furnari and G. M. Farinella. "What Would You Expect? Anticipating Egocentric Actions with Rolling-Unrolling LSTMs and Modality Attention." In: *International Conference on Computer Vision (ICCV)*. 2019.

[80] J. Xu, T. Yao, Y. Zhang, and T. Mei. "Learning Multimodal Attention LSTM Networks for Video Captioning". In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. ACM, 2017, pp. 537–545. ISBN: 978-1-4503-4906-2. DOI: 10.1145/3123266.3123448. URL: http://doi.acm.org/10.1145/3123266.3123448.

[81] C. Hori, T. Hori, T. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. "Attention-Based Multimodal Fusion for Video Description". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4203–4212. DOI: 10.1109/ICCV.2017.450.

[82] X. Yang, P. Ramesh, R. Chitta, S. Madhvanath, E. A. Bernal, and J. Luo. "Deep multimodal representation learning from temporal data". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5447–5455.

[83] H. Tan and M. Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019.

[84] T. Gupta, A. Schwing, and D. Hoiem. "ViCo: Word Embeddings from Visual Co-occurrences". In: *ICCV*. 2019.

[85] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. "VideoBERT: A Joint Model for Video and Language Representation Learning". In: *ICCV 2019* abs/1904.01766 (2019). arXiv: 1904.01766. URL: http://arxiv.org/abs/1904.01766.

[86] D. Wang, P. Cui, M. Ou, and W. Zhu. "Learning Compact Hash Codes for Multimodal Representations Using Orthogonal Deep Structure". In: *IEEE Transactions on Multimedia* 17.9 (2015), pp. 1404–1416. ISSN: 1520-9210. DOI: 10.1109/TMM.2015.2455415.

[87] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. "The Sound of Pixels". In: *The European Conference on Computer Vision (ECCV)*. 2018.

[88] V. Manjunatha, M. Iyyer, J. Boyd-Graber, and L. Davis. "Learning to Color from Language". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 764–769. DOI: 10.18653/v1/N18-2120. URL: https://www.aclweb.org/anthology/N18-2120.

[89] A. Karpathy and L. Fei-Fei. "Deep Visual-Semantic Alignments for Generating Image Descriptions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 664–676. DOI: 10.1109/TPAMI.2016.2598339.

[90] L. Baraldi, C. Grana, and R. Cucchiara. "Hierarchical Boundary-Aware Neural Encoder for Video Captioning". In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 3185–3194. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.339.

[91] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 2048–2057. URL: http://dl.acm.org/citation.cfm?id=3045118.3045336.

[92] J. Lu, C. Xiong, D. Parikh, and R. Socher. "Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3242–3250. DOI: 10.1109/CVPR.2017.345.

[93] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. "Extraction of visual features for lipreading". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002), pp. 198–213. DOI: 10.1109/34.982900.

[94] G. Zhao, M. Barnard, and M. Pietikainen. "Lipreading With Local Spatiotemporal Descriptors". In: *IEEE Transactions on Multimedia* 11.7 (2009), pp. 1254–1265. DOI: 10.1109/TMM.2009.2030637.

[95] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas. "LipNet: End-to-End Sentence-level Lipreading". In: *GPU Technology Conference* (2017). URL: https://github.com/Fengdalu/LipNet-PyTorch.

[96] "Towards Multimodal Sentiment Analysis: Harvesting Opinions from The Web". In: Alicante, Spain.

[97] V. Pérez Rosas, R. Mihalcea, and L. Morency. "Multimodal Sentiment Analysis of Spanish Online Videos". In: *IEEE Intelligent Systems* 28.3 (2013), pp. 38–45. DOI: 10.1109/MIS.2013.9.

[98] S. Poria, E. Cambria, and A. Gelbukh. "Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-level Multimodal Sentiment Analysis". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 2539–2544. DOI: 10.18653/v1/D15-1303. URL: https://www.aclweb.org/anthology/D15-1303.

[99] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria. "Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper)". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Florence, Italy: Association for Computational Linguistics, July 2019.

[100] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. "IEMOCAP: interactive emotional dyadic motion capture database". In: *Language Resources and Evaluation* 42.4 (2008), p. 335. ISSN: 1574-0218. DOI: 10.1007/s10579-008-9076-6. URL: https://doi.org/10.1007/s10579-008-9076-6.

[101] Z. C. Lipton. "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *CoRR* abs/1506.00019 (2015). arXiv: 1506.00019. URL: http://arxiv.org/abs/1506.00019.

[102] T. Wu, T. Chien, C. Chan, C. Hu, and M. Sun. "Anticipating Daily Intention using On-Wrist Motion Triggered Sensing". In: *Intenational Conference on Conmputer Vision* abs/1710.07477 (2017). arXiv: 1710.07477. URL: http://arxiv.org/abs/1710.07477.

[103] E. Aarts and R. Wichert. "Ambient intelligence". In: Jan. 2009, pp. 244–249. DOI: 10.1007/978-3-540-88546-7_47.

[104] K. Ellis, J. Kerr, S. Godbole, G. Lanckriet, D. Wing, and S. Marshall. "A random forest classifier for the prediction of energy expenditure and type of physical activity from wrist and hip accelerometers". In: *Physiological Measurement* 35.11 (2014), pp. 2191–2203. DOI: 10.1088/0967-3334/35/11/2191. URL: https://doi.org/10.1088%2F0967-3334%2F35%2F11%2F2191.

[105] C. Torres-Huitzil and A. Alvarez-Landero. "Accelerometer-Based Human Activity Recognition in Smartphones for Healthcare Services". In: *Mobile Health: A Technology Road Map*. Ed. by S. Adibi. Cham: Springer International Publishing, 2015, pp. 147–169. ISBN: 978-3-319-12817-7. DOI: 10.1007/978-3-319-12817-7-7. URL: https://doi.org/10.1007/978-3-319-12817-7-7.

[106] G. Ogbuabor and R. La. "Human Activity Recognition for Healthcare Using Smartphones". In: *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*. ICMLC 2018. ACM, 2018, pp. 41–46. ISBN: 978-1-4503-6353-2. DOI: 10.1145/3195106.3195157. URL: http://doi.acm.org/10.1145/3195106.3195157.

[107] S. Ji, W. Xu, M. Yang, and K. Yu. "3D Convolutional Neural Networks for Human Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.1 (2013), pp. 221–231. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.59.

[108] Y. Kong, Z. Tao, and Y. Fu. "Deep Sequential Context Networks for Action Prediction". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3662–3670. DOI: 10.1109/CVPR.2017.390.

[109] S. Saguna, A. Zaslavsky, and D. Chakraborty. "Complex Activity Recognition Using Context-driven Activity Theory and Activity Signatures". In: *ACM Trans. Comput.-Hum. Interact.* 20.6 (2013), 32:1–32:34. ISSN: 1073-0516. DOI: 10.1145/2490832. URL: http://doi.acm.org/10.1145/2490832.

[110] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo. "Identifying First-person Camera Wearers in Third-person Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* abs/1704.06340 (2017). arXiv: 1704.06340. URL: http://arxiv.org/abs/1704.06340.

[111] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. P. Singh. "Action-Conditional Video Prediction using Deep Networks in Atari Games". In: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS) (2015). URL: http://arxiv.org/abs/1507.08750.

[112] M. H. Hany El-Ghaish and A. Shoukry. "Human Action Recognition Using A Multi-Modal Hybrid Deep Learning Model". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by G. B. Tae-Kyun Kim Stefanos Zafeiriou and K. Mikolajczyk. BMVA Press, 2017, pp. 84.1–84.13. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.84. URL: https://dx.doi.org/10.5244/C.31.84.

[113] A. Pieropan, G. Salvi, K. Pauwels, and H. Kjellström. "Audio-visual classification and detection of human manipulation actions". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 3045–3052. DOI: 10.1109/IROS.2014.6942983.

[114] J. A. Stork, L. Spinello, J. Silva, and K. O. Arras. "Audio-based human activity recognition using Non-Markovian Ensemble Voting". In: *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. 2012, pp. 509–514. DOI: 10.1109/ROMAN.2012.6343802.

[115] L. Bao and S. S. Intille. "Activity Recognition from User-Annotated Acceleration Data". In: *Pervasive Computing*. Ed. by A. Ferscha and F. Mattern. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–17. ISBN: 978-3-540-24646-6.

[116] A. F. Bobick and J. W. Davis. "The recognition of human movement using temporal templates". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.3 (2001), pp. 257–267. ISSN: 0162-8828. DOI: 10.1109/34.910878.

[117] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. "Actions as space-time shapes". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. 2005, 1395–1402 Vol. 2. DOI: 10.1109/ICCV.2005.28.

[118] Ju Sun, Xiao Wu, Shuicheng Yan, L. Cheong, T. Chua, and Jintao Li. "Hierarchical spatio-temporal context modeling for action recognition". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 2004–2011. DOI: 10.1109/CVPR.2009.5206721.

[119] C. Feichtenhofer, A. Pinz, and R. P. Wildes. "Spatiotemporal Multiplier Networks for Video Action Recognition". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 7445–7454. DOI: 10.1109/CVPR.2017.787.

[120] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "Large-Scale Video Classification with Convolutional Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[121] A. Fathi, A. Farhadi, and J. M. Rehg. "Understanding Egocentric Activities". In: *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 407–414. ISBN: 978-1-4577-1101-5. DOI: 10.1109/ICCV.2011.6126269. URL: http://dx.doi.org/10.1109/ICCV.2011.6126269.

[122] Y. Li, M. Liu, and J. M. Rehg. "In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video". In: *The European Conference on Computer Vision (ECCV)*. 2018.

[123] S. Singh, C. Arora, and C. Jawahar. "Trajectory aligned features for first person action recognition". In: *Pattern Recognition* 62 (2017), pp. 45 –55. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2016.07.031. URL: http://www.sciencedirect.com/science/article/pii/S0031320316301893.

[124] M. Nakanishi, S. Izumi, S. Nagayoshi, H. Sato, H. Kawaguchi, M. Yoshimoto, T. Ando, S. Nakae, C. Usui, T. Aoyama, and S. Tanaka. "Physical activity group classification algorithm using triaxial acceleration and heart rate". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2015, pp. 510–513. DOI: 10.1109/EMBC.2015.7318411.

[125] P. Bharti, D. De, S. Chellappan, and S. K. Das. "HuMAn: Complex Activity Recognition with Multi-Modal Multi-Positional Body Sensing". In: *IEEE Transactions on Mobile Computing* 18.4 (2019), pp. 857–870. ISSN: 1536-1233. DOI: 10.1109/TMC.2018.2841905.

[126] Chun Zhu, Wei Sun, and Weihua Sheng. "Wearable sensors based human intention recognition in smart assisted living systems". In: *2008 International Conference on Information and Automation*. 2008, pp. 954–959. DOI: 10.1109/ICINFA.2008.4608137.

[127] E. A. Bernal, X. Yang, Q. Li, J. Kumar, S. Madhvanath, P. Ramesh, and R. Bala. "Deep Temporal Multimodal Fusion for Medical Procedure Monitoring Using Wearable Sensors". In: *IEEE Transactions on Multimedia* 20.1 (2018), pp. 107–118. DOI: 10.1109/TMM.2017.2726187.

[128] J. Kerr, S. Marshall, S. Godbole, J. Chen, A. Legge, A. Doherty, P. Kelly, M. Smith, H. Badland, and C. Foster. "Using the SenseCam to Improve Classifications of Sedentary Behavior in Free-Living Settings". In: *American journal of preventive medicine* 44 (Mar. 2013), pp. 290–6. DOI: 10.1016/j.amepre.2012.11.004.

[129] G. O'Loughlin, S. Cullen, A. McGoldrick, S. Connor, R. Blain, S. O'Malley, and G. Warrington. "Using a Wearable Camera to Increase the Accuracy of Dietary Analysis". In: *American journal of preventive medicine* 44 (Mar. 2013), pp. 297–301. DOI: 10.1016/j.amepre.2012.11.007.

[130] A. R. Silva, S. Pinho, L. M. Macedo, and C. J. Moulin. "Benefits of SenseCam Review on Neuropsychological Test Performance". In: *American Journal of Preventive Medicine* 44.3 (2013), pp. 302 –307. ISSN: 0749-3797. DOI: https://doi.org/10.1016/j.amepre.2012.11.005. URL: http://www.sciencedirect.com/science/article/pii/S0749379712008501.

[131] W.-C. Cheng and D.-M. Jhan. "Triaxial Accelerometer-Based Fall Detection Method Using a Self-Constructing Cascade-AdaBoost-SVM Classifier". In: *IEEE journal of biomedical and health informatics* 17 (Mar. 2013), pp. 411–9. DOI: 10.1109/JBHI.2012.2237034.

[132] H. Gjoreski, M. Lustrek, and M. Gams. "Accelerometer Placement for Posture Recognition and Fall Detection". In: *2011 Seventh International Conference on Intelligent Environments*. 2011, pp. 47–54. DOI: 10.1109/IE.2011.11.

[133] C. Vondrick, H. Pirsiavash, and A. Torralba. "Anticipating the future by watching unlabeled video". In: *Conference on Computer Vision and Pattern Recognition* abs/1504.08023 (2016). arXiv: 1504.08023. URL: http://arxiv.org/abs/1504.08023.

[134] C. Vondrick, H. Pirsiavash, and A. Torralba. "Anticipating Visual Representations from Unlabeled Video". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 98–106. DOI: 10.1109/CVPR.2016.18.

[135] J. Bütepage, M. J. Black, D. Kragic, and H. Kjellström. "Deep representation learning for human motion prediction and classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* abs/1702.07486 (2017). arXiv: 1702.07486. URL: http://arxiv.org/abs/1702.07486.

[136] M. S. Ryoo, B. Rothrock, and L. H. Matthies. "Pooled Motion Features for First-Person Videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* abs/1412.6505 (2014). arXiv: 1412.6505. URL: http://arxiv.org/abs/1412.6505.

[137] J. Walker, K. Marino, A. Gupta, and M. Hebert. "The Pose Knows: Video Forecasting by Generating Pose Futures". In: *Intenational Conference on Conmputer Vision* abs/1705.00053 (2017). arXiv: 1705.00053. URL: http://arxiv.org/abs/1705.00053.

[138] T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. "Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks". In: *Advances In Neural Information Processing Systems (NIPS)* abs/1607.02586 (2016). arXiv: 1607.02586. URL: http://arxiv.org/abs/1607.02586.

[139] J. Gao, Z. Yang, and R. Nevatia. "RED: Reinforced Encoder-Decoder Networks for Action Anticipation". In: *British Machine Vision Conference* abs/1707.04818 (2017). arXiv: 1707.04818. URL: http://arxiv.org/abs/1707.04818.

[140] S. Ma, L. Sigal, and S. Sclaroff. "Learning Activity Progression in LSTMs for Activity Detection and Early Detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1942–1950. DOI: 10.1109/CVPR.2016.214.

[141] M. Pei, Yunde Jia, and S. Zhu. "Parsing video events with goal inference and intent prediction". In: *2011 International Conference on Computer Vision (ICCV)*. 2011, pp. 487–494. DOI: 10.1109/ICCV.2011.6126279.

[142] A. Bhattacharyya, M. Fritz, and B. Schiele. "Long-Term On-board Prediction of People in Traffic Scenes Under Uncertainty". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), pp. 4194–4202.

[143] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. "Pedestrian Action Anticipation using Contextual Feature Fusion in Stacked RNNs". In: *BMVC*. 2019.

[144] Y. Kong, D. Kit, and Y. Fu. "A Discriminative Model with Multiple Temporal Scales for Action Prediction". In: *European Conference on Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 596–611. ISBN: 978-3-319-10602-1.

[145] M. S. Ryoo. "Human activity prediction: Early recognition of ongoing activities from streaming videos". In: *2011 International Conference on Computer Vision (ICCV)*. 2011, pp. 1036–1043. DOI: 10.1109/ICCV.2011.6126349.

[146] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang. "Recognize Human Activities from Partially Observed Videos". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 2658–2665. DOI: 10.1109/CVPR.2013.343.

[147] M. Hoai and F. De la Torre. "Max-margin early event detectors". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2863–2870. DOI: 10.1109/CVPR.2012.6248012.

[148] K. Li, J. Hu, and Y. Fu. "Modeling Complex Temporal Composition of Actionlets for Activity Prediction". In: *European Conference on Computer Vision – ECCV 2012*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 286–299. ISBN: 978-3-642-33718-5.

[149] F. D. Torre, J. K. Hodgins, J. Montano, and S. Valcarcel. "Detailed Human Data Acquisition of Kitchen Activities: the CMU-Multimodal Activity Database (CMU-MMAC)". In: *CHI 2009 Workshop. Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*. 2009.

[150] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. "The Kinetics Human Action Video Dataset". In: *CoRR* (2017).

[151] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. d. R. Millàn. "Collecting complex activity datasets in highly rich networked sensor environments". In: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 2010, pp. 233–240. DOI: 10.1109/INSS.2010.5573462.

[152] S. Bano, A. Cavallaro, and X. Parra. "Gyro-based Camera-motion Detection in User-generated Videos". In: *Proceedings of the 23rd ACM International Conference on Multimedia*. MM '15. ACM, 2015, pp. 1303–1306. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806336. URL: http://doi.acm.org/10.1145/2733373.2806336.

[153] C. Chen, R. Jafari, and N. Kehtarnavaz. "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor". In: *2015 IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 168–172. DOI: 10.1109/ICIP.2015.7350781.

[154] S. Song, N. Cheung, V. Chandrasekhar, B. Mandal, and J. Lin. "Egocentric Activity Recognition with Multimodal Fisher Vector". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* abs/1601.06603 (2016). arXiv: 1601.06603. URL: http://arxiv.org/abs/1601.06603.

[155] S. Stein and S. J. McKenna. *User-Adaptive Models for Recognizing Food Preparation Activities*. 2013. DOI: 10.1145/2506023.2506031. URL: http://eprints.gla.ac.uk/134990/.

[156] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[157] S. Liu, R. Gao, and P. Freedson. "Computational Methods for Estimating Energy Expenditure in Human Physical Activities." In: *Medicine and science in sports and exercise* 44 (2012), pp. 2138–46.

[158] H. Pirsiavash and D. Ramanan. "Detecting activities of daily living in first-person camera views". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2847–2854. DOI: 10.1109/CVPR.2012.6248010.

[159] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)* abs/1703.10593 (2017). arXiv: 1703.10593. URL: http://arxiv.org/abs/1703.10593.

[160] S. Kiranyaz, T. Ince, and M. Gabbouj. "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks". In: *IEEE Transactions on Biomedical Engineering* 63.3 (2016), pp. 664–675. ISSN: 0018-9294. DOI: 10.1109/TBME.2015.2468589.

[161] S.-M. Lee, S. M. Yoon, and H. Cho. "Human activity recognition from accelerometer data using Convolutional Neural Network". In: *IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2017, pp. 131–134. DOI: 10.1109/BIGCOMP.2017.7881728.

[162] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, and Kong-Pang Pun. "An efficient MFCC extraction method in speech recognition". In: *2006 IEEE International Symposium on Circuits and Systems*. 2006, 4 pp.–. DOI: 10.1109/ISCAS.2006.1692543.

[163] L. Muda, M. Begam, and I. Elamvazuthi. "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques". In: *CoRR* abs/1003.4083 (2010). URL: http://dblp.uni-trier.de/db/journals/corr/corr1003.html.abs-1003-4083.

[164] R. Gonzalez. "Better Than MFCC Audio Classification Features". In: *The Era of Interactive Media*. New York, NY: Springer New York, 2013, pp. 291–301. ISBN: 978-1-4614-3501-3.

[165] L. Jiqing, D. Yuan, H. Jun, Z. Xianyu, and W. Haila. "Sports audio classification based on MFCC and GMM". In: *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*. 2009, pp. 482–485. DOI: 10.1109/ICBNMT.2009.5348520.

[166] O. Ronneberger, P.Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Vol. 9351. LNCS. (available on arXiv:1505.04597 [cs.CV]). Springer, 2015, pp. 234–241. URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

[167] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. "CNN architectures for large-scale audio classification". In: (2017), pp. 131–135. ISSN: 2379-190X. DOI: 10.1109/ICASSP.2017.7952132.

[168] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić. "Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring Using Convolutional Neural Networks". In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. ICMI '17. ACM, 2017, pp. 216–220. ISBN: 978-1-4503-5543-8. DOI: 10.1145/3136755.3136817. URL: http://doi.acm.org/10.1145/3136755.3136817.

[169] H. Ohashi, M. O. A. Al-Naser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, and A. Dengel. "Augmenting Wearable Sensor Data with Physical Constraint for DNN-Based Human-Action Recognition". In: *Time Series Workshop. Time Series Workshop @ ICML, befindet sich ICML 2017, August 11-11, Sydney, Australia*. 2017.

[170] S. Battiato, G. M. Farinella, G. Puglisi, and D. Ravi. "Saliency-Based Selection of Gradient Vector Flow Paths for Content Aware Image Resizing". In: *IEEE Transactions on Image Processing* 23.5 (2014), pp. 2081–2095. ISSN: 1057-7149. DOI: 10.1109/TIP.2014.2312649.

[171] S. Avidan and A. Shamir. "Seam carving for content-aware image resizing". In: *ACM Trans. Graph.* 26.3 (2007), p. 10.