Departamento de Informática

Universidad Técnica Federico Santa María

Proyecto de Tesis

Magíster en Ciencias de la Ingeníeria Informática

Título del Proyecto de Tesis:	"Hybrid CNN+LSTM for Face Recognition in Videos"
Nombre del Alumno:	Bellantonio Marco
Número Teléfono - Celular:	+56 9 6565 1648
E-Mail:	mad.bella88@gmail.com
Fecha de Ingreso al Programa:	1st August 2016
Pregrado:	Bachelor Degree in Computer Engeneering
(Título o Grado, Institución, Año)	University of Bologna, 2014
Profesor Guía de Tesis:	Prof. Ricardo Nañculef, Prof. Sergio Escalera
Fecha Presentación Tema de Tesis:	16th December 2016
Fecha Aprobación Tema de Tesis:	
Fecha tentativa de Término:	
Comisión interna de graduación:	

Contents

RESUMEN

Debe ser suficientemente informativo, y contener una síntesis del proyecto, sus objetivos, resultados esperados y palabras claves. (2 paginas)

El reconocimiento facial, junto con el reconocimiento de las acciones y gestos humanos, son hoy día una de las aplicaciones más exitoso de análisis humana automatizada. Durante los ltimos diez años más o menos, se ha convertido en una zona muy popular de la investigación en computer vision y ha recibido mucha atención por parte de las organizaciones internacionales (Thumos, ChaLearn, etc). [1] Un sistema de reconocimiento facial es una aplicación informática capaz de identificar o verificar una persona a partir de una imagen digital o un fotograma de vídeo. *Verificación* y *identificación* son dos problemas muy distintos en el reconocimiento de rostros. Sistemas de verificación tratan de responder a la pregunta "*Es esta persona la que dice de ser?*". En un sistema de verificación, un individuo presenta a sí mismo como una persona específica, y el problema de verificación se describe generalmente como un pareo 1-a-1, donde un sistema automatizado intenta de hacer coincidir la presencia de un individuo contra una información específica de la misma individuo ya presente en el sistema. Sistemas de identificación, por el contrario, tratan de responder a las preguntas "*Quién es esta persona?*", Y su objetivo es identificar una persona desconocida controlando la información individual contra la que ya están en el sistema de todos los demás. En otras palabras: la identificación es un problema de clasificación multipla descrito como un pareo 1-a-n (donde n es el nmero total de individuo en el sistema), mientras que la verificación es una tarea de clasificación binario con par de vídeos.

En este proyecto se aborda el problema de la identificación de las caras con el uso de un modelo de aprendizaje profundo. Aprendizaje profundo es un campo de aprendizaje de máquinas estrictamente relacionado con la redes neuronales artificiales cuyo intento es modelar abstracción de alto nivel en los datos y aprender varios niveles de la representación mediante la explotación de muchas capas de procesamiento de información no lineal. Está destinado a la extracción de características con o sin supervisión y transformación, para el análisis de patrones y para la clasificación [31] [**?**]. Los recientes avances en análisis facial utilizando marcos de aprendizaje profundas como Redes Neuronales Convolucionales (CNN) o Redes de Creencias Profundas (DBN) proporcionan la idea de realizar composiciones de alta dimensión no lineal [36]. Arquitecturas de aprendizaje profundo han sido ampliamente utilizados en el reconocimiento de rostros [21, 28, 35, 52], en el reconocimiento de expresiones faciales [26, 55] y en la detección des las emociones [23, 25, 36].

Al igual que en muchas otras tareas de computer vision, los datos de entrada para el reconocimiento facial pueden ser muy diferentes, incluyendo imágene, videos, mapas de profundidad [51] [32], imágenes térmicas [50] [39], modelos 3D de la cara [5], etc. Por supuesto, el tipo de datos de entrada plantean diferentes limitaciones y oportunidades a nivel de modelado. Específicamente en los videos, puede ser evidente que la información temporal debe ser explotado para realizar tareas de reconocimiento. De hecho, las obras recientes confirman las ventajas de utilizar modelos temporales como Redes Neuronales Recurrentes (RNN) o Long-Short Term Memory (LSTM) para problemas de análisis de cara humanos, como la detección y seguimiento de los rostros [54], el reconocimiento de la expresión facial [3] y el reconocimiento de emociones [12] [8]. Sin embargo, después de una revisión exhaustiva de las fuentes bibliográficas, llegamos a la conclusión que muy pocos trabajos han abordado el problema del reconocimiento facial usando modelos neuronales temporales y ninguno de ellos se han ocupado de reconocimiento de caras en los videos. En nuestra opinión, esto representa una oportunidad interesante de investigación para contribuciones originales.

En esta tesis se propone de abordar el problema de disear modelos de aprendizaje profundos adaptados para explotar la información temporal contenida en los videos para el reconocimiento de rostros. En concreto, nos proponemos estudiar una arquitectura basada en la CNN-LSTM,

utilizada con éxito para otras tareas de análisis de vídeo como el reconocimiento y la descripción de objetos (image captioning) [11] [48], análisis de sentimiento [49] y clasificación del texto [56], y comparar los resultados obtenidos con otros métodos de reconocimiento facial en estado-of-the-art [17] [52] [44] [7].

Este trabajo se organiza en diferentes fases. En primer lugar, se llevará a cabo una revisión exhaustiva de los más recientes documentos y trabajos en el campo de la visión artificial en relación con los modelos de aprendizaje profundo para el reconocimiento de caras en los videos. En segundo lugar, tenemos la intención de preparar un análisis precisa de los métodos más recientes y eficientes junto con el estudio de los resultados observados y las bases de datos utilizadas. Una vez reunida la información necesaria para estar informado sobre el estado de la técnica, el siguiente paso importante será la definición de las arquitecturas implicadas, Red Neuronal Convolutivas y Long-Short Term Memory, junto con la elección de las bases de datos. La disponibilidad de los datos para el reconocimiento facial de vídeo es grande. La más utilizada base de datos (y también la más difícil) para el reconocimiento facial de vídeo es sin ninguna duda el Youtube Face database (YTF). Sin embargo, en este trabajo se decide construir una nueva base de datos de la conocida base de datos Motion of Body (MoBo). El MoBo DB está destinado a ser utilizado para tareas de detección y reconocimiento de movimientos. Por lo tanto, las imágenes de las que se compone son foto de cuerpo entero de varios temas. En nuestro proyecto aplicamos técnicas de procesamiento de imágenes para detectar el rostro, recortar la región de la cara y almacenar la imagen resultante en un formato adecuado. La nueva base de datos sería una contribución adicional importante de este trabajo.

Después de el diseo de la arquitectura y la elección de las bases de datos, seguirán la aplicación y un conjunto de experimentos.

ABSTRACT (Inglés)

Debe ser suficientemente informativo, y contener una síntesis del proyecto, sus objetivos, resultados esperados y palabras claves. Debe ser equivalente al RESUMEN. (1 pagina)

Face recognition, along with human action and gesture recognition, is nowadays one of the most successful application of automated human analysis. Over the last ten years or so, it has become a very popular area of research in computer vision and has received a lot of attention from international organizations (THUMOS, ChaLearn, etc). [1] A facial recognition system is a computer application capable of identifying or verifying a person from a digital image or a video frame from a video source. *Verification* and *identification* are two very distinct problems in face recognition. Verification systems seek to answer the question "*Is this person who they say they are?*". Under a verification system, an individual presents himself or herself as a specific person, and the verification problem is generally described as a 1-to-1 matching where an automated system tries to match the presence of an individual against a specific information of the same individual already present in the system. Identification systems, on the other hand, seek to answer the questions "Who is this person?", and aim to identify an unknown person by checking the individual information against all others already in the system. In this project, we address the problem of face identification with the use of a deep learning framework. Recent advances in facial analysis using deep learning frameworks such as Convolutional Neural Networks (CNN) or Deep Belief Networks (DBN) provide the notion of realizing non-linear high dimensional compositions [36]. Deep learning architectures have been widely used in face recognition [21, 28, 35, 52], facial expression recognition [26, 55], emotion detection [23, 25, 36].

As in many other computer vision tasks, input data for face recognition can be very different, including raw images, videos, depth maps [51] [32], thermal images [50] [39], 3D face models [5], etc. Of course, the type of input data pose different constraints and opportunities at the modelling level. Specifically in videos, it may be apparent that temporal information should be exploited to perform recognition tasks. Indeed, recent successful works confirm the advantage of using temporal models such as Recurrent Neural Networks (RNN) and Long-Short Term Memory models (LSTM) for human face analysis problems, such as face detection and tracking [54], facial expression recognition [3] and emotion recognition [12] [8]. However, after an intensive literature review, we conclude that very few works have addressed the problem of face recognition using temporal neural models and none of them dealt with face recognition in videos. In our opinion, this represents an interesting research opportunity for original contributions.

In this thesis we propose to address the problem of designing deep learning models tailored to exploit the temporal information contained in videos to perform face recognition. Concretely, we propose to study a CNN-LSTM based architecture successfully used for other video analysis tasks, such as object recognition and description (image captioning) [11] [48], sentimental analysis [49] and text classification [56] to mention few, and to compare the obtained results with other state-of-the-art face recognition methods [17] [52] [44] [7].

This work will be organized in different phases. First of all, an exhaustive review of recent papers and works in the field of computer vision related to deep models for face recognition in videos will be performed. Secondly, we plan to prepare a precise analysis of the most recent and efficient methods along with the study of the performances reported and the databases used. After having gathered the information necessary to be informed and aware of the state of the art, the next important step will be the definition of the architectures involved, namely Convolution Neural Network and Long-Short Term Memory, along with the choice of the databases.

The data availability for video face recognition is big. However, in this work we contribute also by building a novel face database from the well known Motion of Body (MoBo) database.

After the design of the architecture and the choice of the databases, the implementation and a set of experiments would follow.

1 FORMULACIÓN GENERAL DE LA PROBLEMÁTICA Y PROPUESTA DE TESIS

Debe contener la exposición general del problema, identificando claramente qué aspectos relacionados con la informática son los más relevantes. Además, deberá contener el marco teórico, la discusión bibliográfica con sus referencias y, finalmente, su propuesta de tesis.
(La extensión máxima de esta sección es de hasta 5 páginas. En hojas adicionales incluya la lista de referencias bibliográficas citadas)

1.1 Introduction

Accurately identifying people has always been a very human process. It is a task that we perform routinely and effortlessly in our daily lives. In the past 30 years, the wide availability of powerful and low-cost computers, as well as the development of high-performing embedded computing systems, have aroused an enormous interest in automatic processing of digital images in a variety of applications, including human-computer interaction, surveillance, biometric authentication, multimedia management, and so on and so forth. Research and development in automatic face recognition have followed naturally.
As one of the most successful applications of image analysis and understanding, face recognition has recently gained significant attention. Over the last ten years or so, it has become a very popular area of research in computer vision and one of the most successful applications of human analysis. Nowadays, recent technologies i.e. the Facebook AI Lab FR systems are able to recognize face with an incredible accuracy of more than 97%, which is outstanding, but still less accurate than a human. As a matter of fact, generally speaking computers have always been more accurate than us. But when we deal with artificial intelligence tasks, especially those which involve visual processes such as face/action/gesture recognition, humans capabilities are truly challenging to outperform. The human ability to identify people and objects by sight is without any doubt its most developed form of identification. Despite many different theories on the functioning of the human brain, what is easily understood is its ability to visually recognize recurrent pattern. This idea is behind the inspiration of one of the most widely used and successful method for image processing, namely Convolutional Neural Network.
Convolutional Neural Networks (CNN) are biologically-inspired variants of Multi Layer Perceptrons proposed by Yann LeCun in 1998 [27]. Inspired by the biological functioning of the visual system, CNNs exploit spatially-local correlation by enforcing local connectivity pattern between units (neurons) of adjacent layers. In CNN each filter is replicated and locally shares the parametrization (weights). Figure 1 shows a standard CNN architecture.

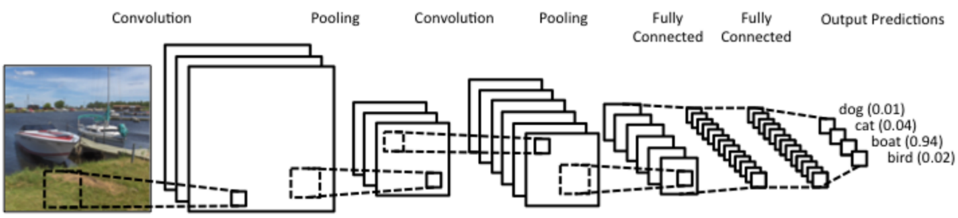


Figure 1: CNN architecture

From figure 1 we can see that feature maps are obtained by repeated application of the filter function across sub-regions of the entire image. From a procedural point of view, CNN outputs (heatmaps) are obtained by "convolving" the input (images or previous layers) with a linear filter, adding a bias term and then applying a non-linear function.
In practice, suppose that we have a $N \times N$ input image, if we use an $m \times m$ filter ω, our first

convolutional layer output will be of size $(N - m + 1) \times (N - m + 1)$. Input pixels x_{ij} are multiplied by the filter components (weights ω) and summed up. Finally, a non-linear function σ is applied. A formalization of this process is shown in equation 1 underneath:

$$y_{ij} = \sigma \left(\sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} x_{(i+a)(j+b)} \right) \tag{1}$$

In the field of computer vision, CNNs are fed with input images and have units (neurons) arranged in 3 dimensions, respectively width, height and depth. In order to extend CNNs to the video domain and to capture temporal information, the most widely used approach consist in extending the convolution along the temporal axis in what is well known as a 3D Convolutional neural network. 3D convolution captures discriminative features along both spatial and temporal dimensions and nowadays most of the methods for human behaviour analysis where temporal information is available.
Generally speaking, one of the most used network for temporal analysis is Recurrent Neural Network (RNN). RNNs can take into account the temporal information by using recurrent connections in hidden layers and can deal with sequences of variable length by defining a *recurrent relation* over timesteps according to the formula

$$S_t = f(W_x X_t + W_r S_{t-1}) \tag{2}$$

where S_t and X_t are respectively the state and the input at time t, S_{t-1} represents the state at the previous timestep, W_r is the so-called transition matrix and W_x are the weights parameters in feed-forward networks. The weights matrices W_x and W_r are filters that determine how much importance both the present input and the past hidden state have. The final output of the network Y_t at a certain time step t is typically computer from one or more states $S_{t-1}..S_{t+1}$. The cost function of the network (e.g. Mean Squared Error) is generally computed over all sequences of the input data and Recurrent Neural Network are generally trained using the Back Propagation Through Time algorithm. BPTT is derived from the basic Back Propagation algorithm, and it differs only because the recurrent network needs to be unfolded through time for a certain amount of time steps. The backward step will begin with computing the gradient of the cost function ξ with respect to the output of the network y. The gradient $\frac{\partial \xi}{\partial y}$ will then be propagated backwards through time (layer by layer) from output to input to update the parameters (weights). Formula 3 shows a formalization of the gradient propagation.

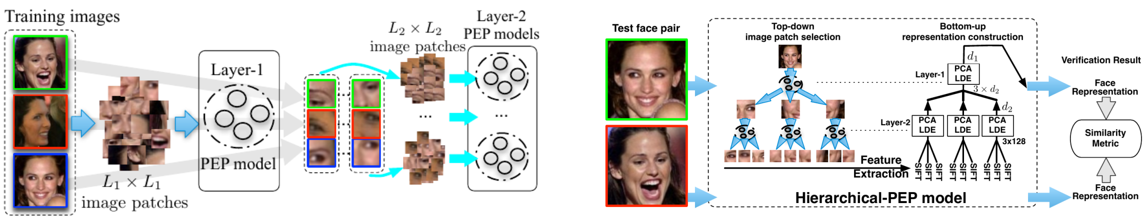
$$\frac{\partial \xi}{\partial S_{t-1}} = \frac{\partial \xi}{\partial S_t} \cdot \frac{\partial S_t}{\partial S_{t-1}} = \frac{\partial \xi}{\partial S_t} \cdot w_r \tag{3}$$

The recent revived interest on RNN is mainly attributed to its recent success in many practical applications such as language modeling [37], speech recognition [4] [16], machine translation [45] [22] and conversation modeling [40], to name a few. But although RNN can be trained through time to learn and memorize what happened in the past, they are characterized by a very short-term memory, which is insufficient for real long-term world applications. In this sense, there were some major mathematical difficulties identified by Hochreiter [18] and Bengio [6] while training RNNs. In both work they demonstrate that basic gradient-based methods appear inadequate for recurrent network which have to learn long range input/output dependencies. To solve this problem (as well as the problem of the vanishine or exploding of the gradient) Long Short-Term Memory (LSTM) [13] was proposed.
LSTMs are particular implementation of Recurrent Neural Network proposed in 1997 by German researchers S. Hochreiter and J. Schmidhuber, usually used as a hidden layer of RNN. But unlike most RNNs, LSTM networks are well-suited to learn to classify, process ad predict time series from very long time windows (up to 1000 discrete time steps), generally of unknown size. LSTMs contain information outside the normal flow of the recurrent network in a gated cell.

Information can be stored in, written to, or read from a cell, similarly to how data are treated in a computers memory. The cell makes decisions about what to store and when to allow reads, writes and erasures, via gates that open and close. Those gates are called input gate, forget gate and output gate.
This work aims to exploit temporal information contained in consecutive video frames by using a LSTM. More specifically, the input of the LSTM network are not directly video frames, rather features extracted using a CNN. This process would hopefully lead to good results in a face recognition problem and also improve the performances of the CNN alone, providing an interesting case study for future extensions.

1.2 Deep Learning Methods for Video Face Recognition

This section examines in more detail how deep learning methods have been used to address the challenges of face recognition, spanning from image quality to unconstrained scenario, from change in pose to occlusions.
Li et al. [29] propose a deep hierarchical version of the PEP model, called Hierarchical Probabilistic Elastic Part-Model, to approach unconstrained face recognition problems. In order to build pose-invariant part-based face representations, faces are decomposed into parts using PEP model hierarchically. From top-down in the hierarchy, the H-PEP model builds pose-invariant face representation for both images and videos. Following in the hierarchy from bottom-up, face part representations are stacked at each layer. By aggregating FPR layer by layer, the method is able to build compact and pose invariant face representations. Figure 2 shows the process (subfigure 2a) and the face representation construction steps (subfigure 2b) of the H-PEP workflow.



(a) 2-layer H-PEP training process

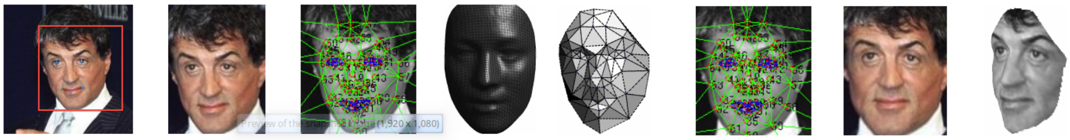
(b) H-PEP construction of face representation

Figure 2: Hierarchical-PEP workflow

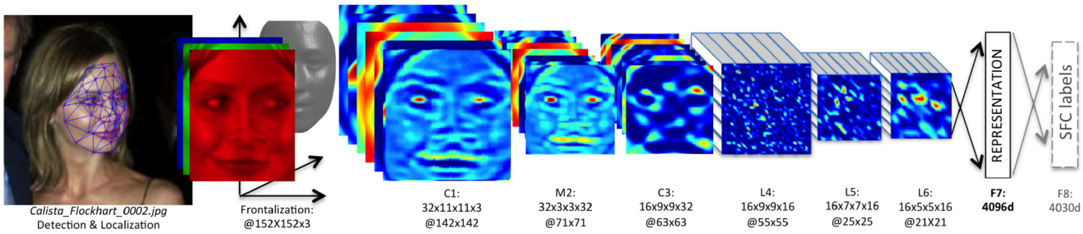
In 2014 Goswami et al. [15] presented a memorability based frame selection algorithm that enables automatic selection of memorable frames for facial feature extraction and matching. A deep learning algorithm was proposed to utilizes a stack of denoising Autoencoders and deep Boltzmann Machines and perform face recognition using the most memorable frames. This work provided the idea to use Autoencoders in order to perform dimensionality reduction of the method presented in this work. Further details will be presented in section 4.6.

CNN-based Methods

Many recent studies have reported the success of using deep CNN in face related tasks. The already cited work by Taigman et al. [46] called DeepFace is based on a very deep CNN architecture together with an alignment technique. The authors revisit face alignment and representation by employing explicit 3D face modeling in order to apply piece-wire affine transformation and derive a face representation from a 9-layer deep neural network. Particularity of the network is that it involves more than 120M parameters using several locally connected layers without shared weights, rather that the standard convolutional neural network. Figure 3a illustrate the alignment pipeline process, whereas figure 3b shows the 9-layer architecture.



(a) Alignment pipeline



(b) A architecture

Figure 3: DeepFace

Inspired by GoogLeNet, Sun et al. [42] used a very deep CNN network with multiple levels of supervision, called Deep hidden IDentity features (DeepID), which reaches human-level face verification performance in the LFW dataset by achieving 97.45% accuracy. DeepID features are built on top of the feature extraction hierarchy of a deep CNN (last hidden layer neuron activations). The proposed features are extracted from various face regions to form complementary and over-complete representations. Recently in 2016, Yang et al. [53] presented a Neural Aggregation Network (NAN) for video face recognition which takes a face video or face image set of a person with variable number of face frames as its input, and produces a compact and fixed dimension visual representation of that person. The whole network is composed of two modules. The feature embedding module is a CNN which maps each face frame into a feature representation. The neural aggregation module is composed of two content-based attention blocks which are driven by a memory storing all the features extracted from the face video through the feature embedding module. The output of the first attention block adapts the second, whose output is adopted as the aggregated representation of the video faces. Due to the attention mechanism, this representation is invariant to the order of the face frames. Important the work of Parkhi et al. [35], in which they made two important contributions: first, they designed a procedure to assemble a large scale dataset; secondly, they trained a deep CNN achieving results comparable to SOTA methods. Sun et al. in 2015 [44] proposed to learn high-performance deep ConvNets with sparse neural connections called Sparse ConvNets. Sparse ConvNets are learned in an iterative way, each time one additional layer is "sparsified" and the entire model is re-trained given the initial weights learned in previous iterations. Important novelty is a new neural correlation-based weight selection criterion which empirically verifies its effectiveness in selecting informative connections from previously learned models at each iteration.

Temporal Deep Learning Models

Among many variants of RNNs, Long Short-Term Memory (LSTM) is arguably one of the most widely used. Other than supervised learning, LSTM is also used in recent work in image generation [47] [17], demonstrating its capability of modeling statistical dependencies of imagery data. LSTM are also widely applied to time series prediction, speech and handwriting recognition, music composition and human action recognition. In the literature we can find few methods which use RNN and LSTM for many different problems involving human faces.

Yoo et al. [54] presented a new robust algorithm that improved face detection and tracking in video sequences by using geometrical facial information and a recurrent neural verifier. In particular they defined a new method called *Three-Face Reference Model* (TFRM) which brings the advantage of a better match process. Other authors such as Graves et al. [3] proposed a new approach for facial expression recognition which combines state-of-the-art techniques for model-based image interpretation and sequence labeling. The Candide-3 face model is used in conjunction with a learned objective function for face model fitting, therefore the resulting sequence of model parameters is presented to a Long-Short Term Memory for classification. The classification algorithm is explicitly designed to consider sequences of data as well as the temporal dynamics of facial expressions. Recently in 2016, Ebrahimi et al. [12] proposed an hybrid CNN-RNN architecture for facial expression analysis and emotion recognition in videos. The authors assert that spatio-temporal evolution of facial features is one of the strongest cues for emotion recognition. The proposed approach uses temporal averaging for aggregation and outperforms other modalities. Again in 2016, Chao et al. [8] present a multi-modal (Audio-visual-physiology) approach to dimensional emotion recognition with a LSTM-RNN architecture. In their work they investigate ϵ-insensitive loss function (instead of squared loss) and temporal pooling. From their work we know that ϵ-insensitive loss function is more robust to label noise and can ignore small errors to get stronger correlation between predictions and labels.

From this brief review we can notice that RNN and LSTM have been used for some human face analysis tasks. Nonetheless, only few methods faced the problem of face recognition using temporal models. We are aware from Corrêa et al. [10] that in face classification a LSTM can be very useful to reduce the number of training samples as well as training time. They also compared the performances of a LSTM model with a standard Multi Layer Perceptron (MLP) in standard face classification problems. From their experiments, LSTM presented better performance in terms of training time, mean square error and correct classification rate. Today, RNNs and LSTMs are an important part of the deep model toolkit for sequence modeling tasks, including human action recognition. However, to the best of my knowledge, there is a lack of methods which use LSTM networks to perform face recognition in videos. Motivated by the lack of related methods, we decided to focus our work on this architecture, with the specific goal of understanding and investigating whether it can improve a CNN-based model reaching state-of-the-art performances. This work is also inspired by some of the aforementioned methods [54] [3] [12] which proposed to use RNN or LSTM as extensions to other deep method.

To summarize what has been said so far, table 1 presents the most recent deep models for video face recognition along with the study of the performances reported and the databases used.

Table 1: Summary of the most recent SOTA works in video face recognition

Work	**Year**	**Database**	**Accuracy**
DeepID2+	2014	LFW	99.47% (95%*)
		YTF	93.2
DeepFace	2014	LFW	97.35%
		YTF	91.4%
H-PEP	2015	LFW	91.1%
		YTF	87%
Sparse ConvNet	2015	LFW	99.55% (96.2%*)
		YTF	93.5%
NAN	2016	YTF	95.72%

* Identification, all others are for verification

1.3 Proposed Method

We propose a combination of CNN and RNN for a hybrid framework to exploit both spatial and temporal information of face features for video face recognition. The system presented in this work is defined in the following.
Each input image X_i is a $N \times N$ pixel's matrix. The Convolutional Neural Network is fed with input images and for each image produces an output feature vector f_i, extracted from one of the last fully connected layer. Figure 4 shows a sketch of a CNN architecture (fed with video frames) and the extraction of the feature vector.

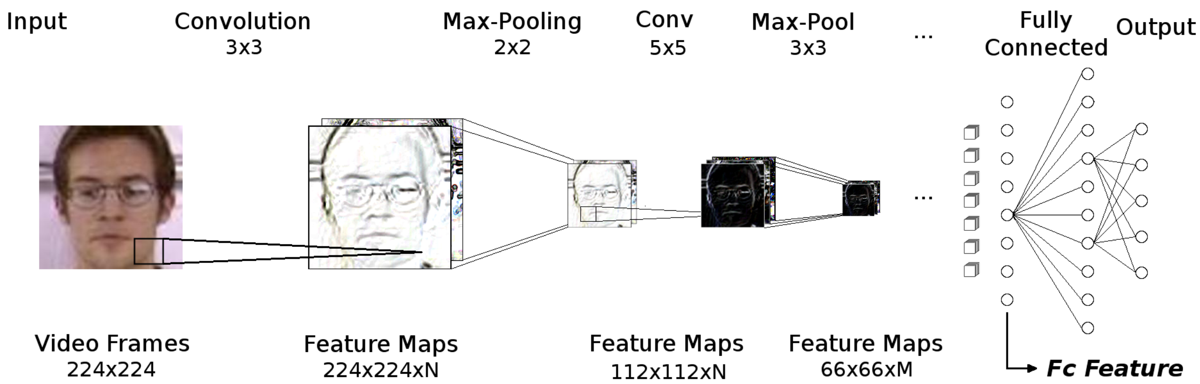


Figure 4: CNN Sketch

From formula 1 we know the output y_{ij} of a convolutional layer. Formula 4 represents the output o_j a last fully connected layer:

$$o_j = \sigma\left(\sum_i \omega_{ij} y_i\right) \tag{4}$$

where j is the number of hidden units contained in the fully connected hidden layer, σ is the non-linear activation function of each neuron and y_i is the i-output of the previous convolutional (or pooling) layer. In the VGG-16 network, for instance, the layer usually used for feature extraction is the 7^{th} fully-connected layer, called *fc7*. Feature vectors f_j from the last fully connected layer would be input of the Long-Short Term Memory (LSTM) Network. In the LSTM, the labels are predicted sequence-wise, *i.e.* given a sequence of n frames $X_i \in \{X_1, ..., X_n\}$, the target prediction is the face identity of the X_n frame. Thus, training is set so that the information contained in the past frames is used in order to predict the current pain level. The temporal window defines the number of consecutive frames that have to be taken into account when predicting a target frame. Therefore the output of the LSTM is the last frame of the defined temporal window. Figure 5 shows a sketch of the designed system. In this case the temporal window is N frames. It is important to notice that the prediction is performed *only* on the last (N^{th}) frame of the input sequence, whereas the previous $N-1$ frames are automatically ignore by the system.
The basic LSTM model, originally proposed by Hochreiter and Schmidhuber [19], is called Vanilla LSTM. As obvious, in the literature we can find different versions of LSTM, accordingly defined for specific needs. One popular variation, introduced by Gers & Schmidhuber in 2000 [14], is built by adding "peephole connections", allowing the gate layers to look at the cell state. Otte et al. (2014) [34] improved the convergence speed of the LSTM by adding recurrent connections between the gates of a single block (but not between the blocks) in what they call a Dynamic Cortex Memory (DCM). Always in 2014, Sak et al. [38] introduced a linear projection layer that projects the output of the LSTM layer down before recurrent and forward connections

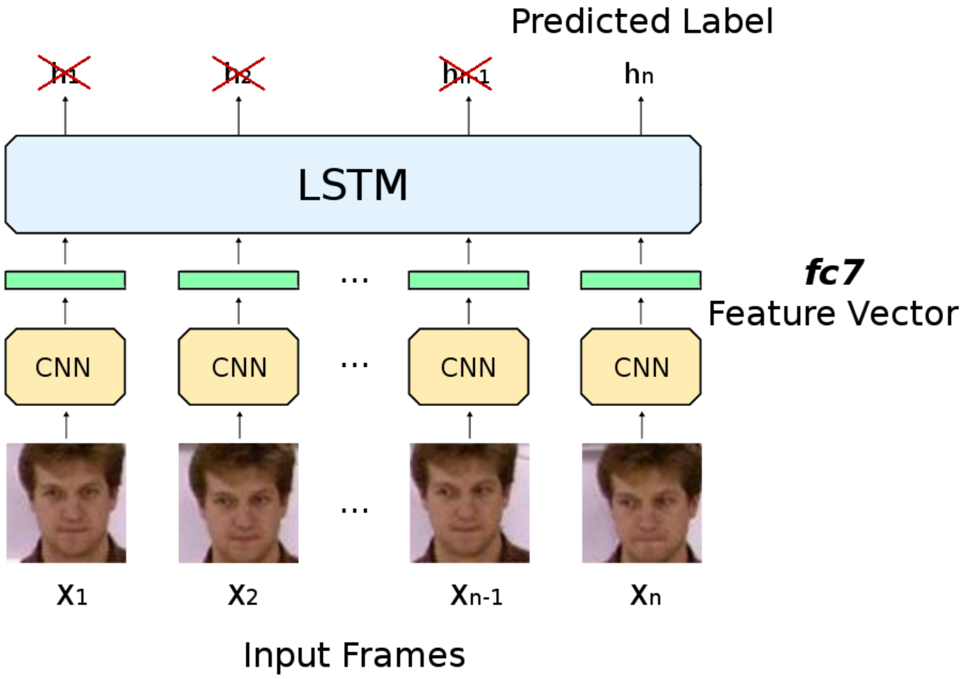
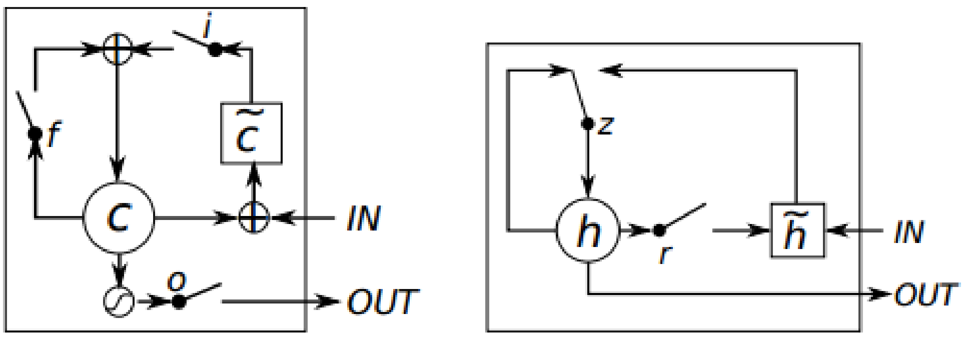


Figure 5: CNN+LSTM system

in order to reduce the amount of parameters for LSTM networks with many blocks. A more drastic variation of the basic LSTM is the Gated Recurrent Unit (GRU) introduced by Cho, et al. (2014) [9]. This model combines the forget and input gates into a single update gate, and it also merges the cell state and hidden state, with some other minor changes. The resulting model is simpler than standard LSTM models and its popularity has been growing increasingly in these past two years. Figure 6 shows the main differences between a LSTM block and a GRU block.



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 6: CNN+LSTM system

As we can see, the input gate i, forget gate f and output gate o present in the LSTM are replaced by the reset gate r and the update gate z in the GRU block.
It is also relevant to mention other few notable LSTM variants, such as Depth Gated RNNs by Yao, et al. (2015) and Clockwork RNNs by Koutnik, et al. (2014).
In order to understand the differences in such a great number of RNN variations, Greff, et al. (2015) did an exhaustive comparison of the most popular ones, finding that they are all about the same. Also Jozefowicz, et al. (2015) tested many variation of RNN architectures, finding that some work better than LSTMs on certain tasks.

From both researches we can conclude that:

1. The basic Vanilla LSTM is generally more efficient than any normal RNNs.
2. Other variations, built for specific problems, are not worth for our ojective.
3. Dropout is necessary and it often improves performances.
4. Learning rate and network size are the most crucial tunable LSTM hyperparameters.

All the LSTM parameters, along with the CNN settings, would be defined later and would be fine-tuned to get the best performances. Implementation details are provided in section 4.

2 HIPÓTESIS DE TRABAJO

Formule las hipótesis de trabajo señalando claramente su conjetura. (1 pagina)

The scenario of my problem is Face Identification in Videos. The system is trained with video (or frame sequences) in constrained environment and tested against the same type of data. The goal of this work is to design a face recognition system based on a deep convolutional neural network and a recurrent neural network. In addition we investigate the capabilities of a Long-Short Term Memory network in improving the performances of a deep learning cnn-based model. As we already know, CNN-based models are very performing for many image recognition. When videos are available, CNN are extended with different strategies, especially those which consider temporal information. The aim of this work is to examine whether a LSTM would be able to exploit temporal information present in consecutive video frames, in particular by capturing dependencies in features of consecutive frames. Hopefully, those dependencies can be exploited to boost the performances of an existing model, in my case a Convolutional neural network, creating an hybrid system capable of reach state-of-the-art results.

3 OBJETIVOS

3.1 Objetivos Generales

The generic objective of my work is to investigate whether an Long-Short Term Memory Network can improve the performances of a CNN-based model in Video Face Recognition problems. In particular:

1. To improve accuracy of a CNN-based deep learning method for face recognition in videos.
2. To build a new public available framework for video face recognition.
3. To compare the outcomes of the plain CNN with the CNN+LSTM system in order to investigate how temporal information affects the performances.

3.2 Objetivos Específicos

The specific objectives which lead my work are:

1. To research the most relevant and up-to-date works concerning deep learning models for video face recognition.
2. To design, implement and assess a novel architecture based on Convolutional Neural Network and Recurrent Neural Networks for video face recognition.
3. To choose a development environment and a programming language to implement the proposed system.
4. To determine which databases are available to assess video face recognition algorithms and which of them are more relevant for the purpose of this project.
5. To propose a procedure to train the presented system.
6. To compare the proposed model with state-of-the-art results in video face recognition.
7. To assess different neural models for learning from sequences, including Simple RNN, LSTM and GRU, both in terms of accuracy and training time.
8. To obtain the accuracy when using a linear classifier on the plain CNN features against the LSTM predictions in order to understand how the use of temporal information can affects the CNN outcomes.
9. To propose a methodology to determine the size of the temporal window used to train recurrent models.

4 METODOLOGÍA Y PLAN DE TRABAJO

After an exhaustive research of the most recent papers and works on deep learning based method for face recognition, the next step is a clear definition of the methodology. From the previous sections we can gather all the most relevant information about the architectures and the databases related to face recognition.

4.1 Video Databases for Face Recognition

In the literature we can find several databases for face recognition problems. Table 2 illustrate the main characteristics of the most used databases. For each database we report the year, the modality, some details such as number of videos and subjects present, and the evaluation strategy (or metric) suggested by the authors of the database.

Table 2: Face Video Databases

Database	**Year**	**Modalities**	**Details**	**Evaluation Metric**
Celebrity 1000 (C1000)	2014	RGBv, face region, facial landmark	159726 videos 1000 subjects	os/cs protocol
Chokepoint	2011	RGBv, RGBi	48 videos 54 subjects	V2V
CMU Motion of Body (MoBo)	2001	RGBi, RGBv	600 videos 24 subjects	-
COX Face	2015	RGBi, RGBv	3000 videos 1000 subjects	V2V, V2S, S2V
Honda/UCSD	2005	B/W videos	75 videos 20 subjects	-
MOBIO	2010	Audio, RGBv	1824 a/v 152 subjects	-
PaSC	2013	RGBi, RGBv	2802 videos 293 subjects	S2S, V2V, S2V
UNBC-McMaster Shoulder Pain	2011	RGBv, FACs, AAMs	200 videos 25 subjects	S2S, V2V, S2V
vidTIMIT	2003	Audio, RGBv	430 a/v 43 subjects	-
WebV-Cele	2009	RGBv, coord, SIFT, CH	75073 videos 2427 subjects	-
YouTube Celebrities	2008	RGBv, BB	1910 videos 47 subjects	-
YouTube Face Dataset (YTF)	2011	RGBv Hand Pos	3425 videos 1595 subjects	10-fold CV Pair-Match

Notes: a/v: audio/video, os/cs: open-set/close-set, V: Video, S: Still image, CV: cross-validation

Some of the databases showed in table 2 are made for various aims: from algorithm's robustness in a real-world scenario to the capability of handling occlusions. Nevertheless, there exist other databases for face recognition such as Labeled Faces in the Wild (LFW), IARPA Janus Benchmark A(IJB-A), PaSC, Oxford Buffy db, ScFace, CMU-FIA, CameFace, Face96, MBGC, ND-Flip-QO, UMD ComCast10, ESOGU Face Videos , MAHNOB-HCI, MMSE-HR and Trailed Face Dataset. Most of the aforementioned databases do not contain videos or are defined for specific problems. For this reason they are not suitable for my task.

4.2 Chosen Datasets

CMU Motion of Body (MoBo) Database

The MoBo database contains 25 individuals walking on a treadmill in the CMU 3D room. The subjects perform 4 different activities: slow walk, fast walk, incline walk and walking with a ball. All subjects are captured using 6 high resolution color cameras distributed evenly around the treadmill. The database contains a total of 600 videos, 340 frames each makes 204,000 video frames. The dataset is challenging for its profile and semi-lateral camera views, where the face is partially visible due to the tilt of the head. In order to evaluate and fairly compare the proposed model, we gather all the papers which use the MoBo data set for face recognition. In table 3, for each method we report the reference, the face region (if extracted), the splits of the database used to evaluate the method and the accuracy along with the specific metric. The protocol follows always the same idea: one activity for training and the remaining three activities for testing. Only in one method the authors split the database into 2 subset without taking into account the number of activities contained in the split.

Table 3: MoBo methods

Paper	Face Region	Protocol	Accuracy
Towards Large-Scale Face Recognition Based on Videos	-	1 train / 3 test	98.1% (CR)
Learning Personal Specific Facial Dynamics for Face Recognition From Videos	40x40	$\frac{1}{2}$ train / $\frac{1}{2}$ test	97.9%
Joint sparse representation for video-based face recognition	30x30	1 train / 3 test	96.5% (IR)
Face Recognition Based on Image Sets	40x40	1 train / 3 test	95.3, 98.1(CR)
From Still Image to Video-Based Face Recognition: An Experimental Analysis	40x40	1 train / 3 test	92.3% (RR)

Notes: RR: Recognition Rate, IR: Identification Rate, CR: Classification Rat

The Motion of body database was meant to be used for motion detection and recognition problems, thus it contains full body pictures of the subjects. In order to extract the face region from each frame, a pre-processing step is necessary. Mobo DB pre-processing will be presented in section 4.3.

YouTube Face (YTF) Database

YoutTube Face is a database of face videos designed for studying the problem of unconstrained face recognition in videos. It contains 3425 videos of 1595 people (average of 2.15 videos for each subject). Considering that the video clip lengths vary from 48 to 6070 frames (average of 181.3 frames/video), we have approximately 620,000 frames.
From the formal definition, YTF is a *verification* dataset. The standard verification protocol from main reference is described as follow:

1. Randomly collect 5000 videos pairs, half are pairs of videos of the same person, half of different people.

2. Pairs are divided into 10 splits. Each split contains 250 same and 250 not-same pairs. Pairs are divided ensuring that the split is subject-mutually exclusive. Subject appears in one split does not appear in anyone else.
3. 9 splits for training and 1 for testing.

The Youtube Face Database contains a large number of subjects and the actions performed are naturally varied (as opposed to performing prescribed actions). It is easier to acquire, thus allowing the baselines to be used by the research community at large. All subjects also have still images available in the Labeled Faces in the Wild (LFW) database [20], thus allowing baselines to be compared to the video to still image matching scenario. The main challenging part is the low image quality: frames sequences of YouTube videos are generally worse than web photos, mainly because of motion blur or viewing distance.
As for the MoBo database, we collected in table 4 the methods which use YTF to perform face recognition. In the table we report the work, the protocol used, the evaluation metric used to evaluate their method along with the obtained results.

Table 4: YTF methods

Paper	**Protocol**	**Metric**	**Result**
DeepID2+ [43]	Standard protocol	ACC	93.2% (VR) 95% (IR)
DeepFace [46]	Standard protocol (unrestricted)	ACC	91.4% (CR)
		100%-EER	92.5%
Eigen-PEP for video face recognition [30]	Standard protocol	ACC	85.4%
Face Recognition in Movie Trailers via Mean Square Sparse Representation-based Classification [33]	Standard protocol	ACC	75.3%,
		AUC	82.9%
		EER	25.3%
Hierarchical-PEP model for real-world face recognition [28]	Not specifically defined	ACC	87%
MDLFace [15]	3M face images of 50K identities	ACC	97.9%
Neural Aggregations Networks [52]	100 frames for each video	ACC	96.5% (IR)
		AUC	98.7%
Sparsifying Neural Network Connections [44]	Train: 290K faces; Val: 47K faces; Test: 5K pairs of faces	ACC	93.5% (RR)
Unconstrained Face Recognition [7]	Own gallery (YTF+LFW) + fusion	ACC	**79%**

Notes: ACC: Accuracy, AUC: area under the curve, EER: Equal Error Rate, RR: Recognition Rate, IR: Identification VR: Verification , Rate, CR: Classification Rate

UNBC-McMaster Shoulder Pain Expression Archive Database

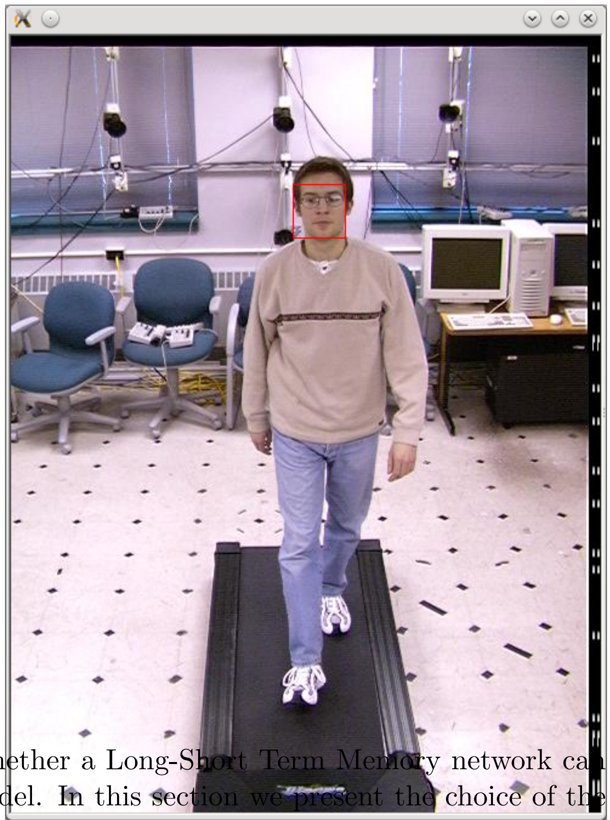
UNBC-McMaster is a pain expression database collected by researchers at McMaster University and University of Northern British Columbia. The database contains facial video sequences of participants who had been suffering from shoulder pain and were performing a series of active and passive range of motion tests to their affected and unaffected limbs on multiple occasions. The database was originally created by capturing facial videos from 129 participants (63 males and 66 females). The participant had a wide variety of occupations and ages. During data capturing the participants underwent eight standard range-of-motion tests: abduction, flexion, and internal and external rotation of each arm. At present, the UNBC-McMaster database contains 200 video sequences of 25 subjects. As the description suggests, the database was thought for pain detection or estimation, therefore it is really challenging because of the changing of the face expression due to the shoulder pain. Additionally, it also provides enough materials to perform face recognition. In our work it will be used as a additional dataset, given that in the literature there are no methods which use it to perform face recognition. If successful, we may consider this work as a baseline for future comparison.

4.3 Image Pre-processing

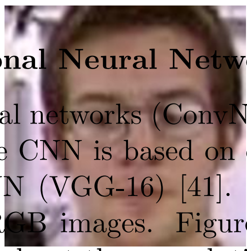
In order to adapt the Motion of Body (MoBo) dataset to our problem, a pre-processing phase is necessary. From each frame the face is detected using a state-of-the-art face detector. Moreover, the face is cropped using the relative coordinates of the detected face. In some cases the face detector fails due to the tilt of the head. In those cases the face region is interpolate from the previuos frame. Each cropped face region is finally saved as new JPG image.
The face detector (available at http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html) is called *dlib* Real-Time Face Pose Estimation, implementation of an excellent paper from the 2014 CVPR Converence [24].
In the following we will show an example of face detection and face region extraction, with the final storing of the resulting image.

```
> python face_detector.py im02_19451807.jpg

processing file:  im02_19451807.jpg
number of faces detected:  1
detection position left,top,right,bottom:
232 122 275 166
```



```
> convert im02_01444804.jpg -crop $position
-resize 224x224 im02_01444804_cropped.jpg
```



4.4 Architecture

The objective of this work is to understand whether a Long-Short Term Memory network can improve the performances of a CNN-based model. In this section we present the choice of the two architectures and the implementation details.

Convolutional Neural Network

Convolutional networks (ConvNets) currently set the state of the art in visual recognition. The design of the CNN is based on one of the recent best-performing models , namely VGG-Very-Deep-16 CNN (VGG-16) [41]. From its formal definition, the VGG-16 inputs are fixed-size 224×224 RGB images. Figure 7 illustrates the VGG-16 network architecture, with precise information about the convolutional and pooling layers.
Here we can notice that the input images have to be of size 244×244, and that the last fully connected layer has a dimensionality of 4096 elements. Therefore the input of the LSTM would be a 1×4096 vector.

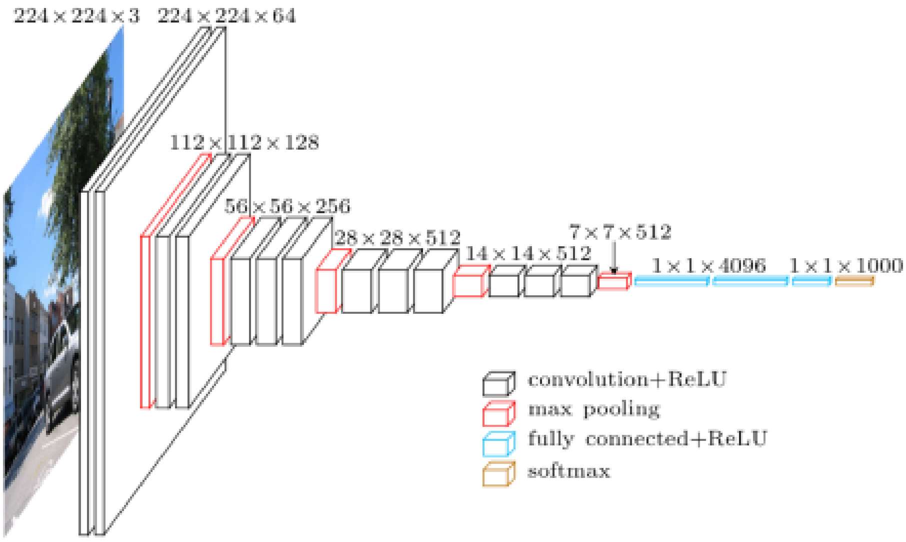


Figure 7: VGG-16 CNN

For this work we chose a pre-trained model, trained from scratch using 2.6 Million images of celebrities collected from the web. The CNN descriptors are computed as described in [35]. For our purpose we need to use the CNN fully connected layer as input for a LSTM, therefore no further modification of the CNN classifier is necessary.

CNN Implementation Details

The code and the VGG-16 pre-trained model is publicly available from the University of Toronto website [2]. The main reference offers the face descriptor source code and the models for Matlab, Torch and Caffe. Using Caffe, the code to obtain the output of a pre-trained mode is really straightforward. In order to read the deploy file and the already precomputed weights, caffe offers the function `caffe.Net(model, weights)`. The network is fed with each image and a forward step is performed:

```
net.blobs['data'].data[...]  = transformer.preprocess('data', img)
out = net.forward()
```

The output could be finally stored in a HDF5 file with

```
outputs.append(h5py.File(outputFile + '.h5', 'w'))
```

Long-Short Term Memory

As already presented in the section 1.3, the LSTM used in this project is one-layer Basic Vanilla LSTM. Figure 8 show a sketch of a LSTM architecture.

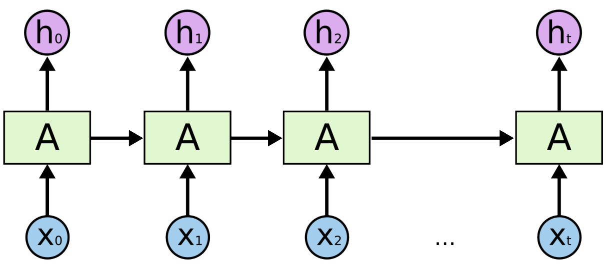


Figure 8: LSTM Architecture

We already know from the CNN definition that the input of the LSTM are 4096×1 vector.

LSTM Implementation Details

There are several possible frameworks for the implementation of the LSTM. The most used and known are Caffe, Keras, Lasagne, TensorFlow, Theano, Torch. A detailed research and final comparison of the best and most efficient frameworks for the desing of the LSTM is important.
Table 5 shows a detailed comparison of the five most used frameworks for deep learning. For each framework we report the base language, the GPU support availability, the recurrent neural network design-ability and the compilation time efficiency.

Table 5: Frameworks comparison

Framework	**API**	**GPU**	**RNN fit**	**Compile time**
Theano	Python+Numpy	Yes*	Good	Slow for large models
Torch	Lua	Yes	Not good	Acceptable
TensorFlow	Python & C++	Yes	Good	Slow
Caffe	Python & C++	Yes	Not good	Slow
Keras	Python	Yes	Good	Acceptable

* No multi-gpu by default

4.5 Fine-tuning CaffeNet pretrained model

In order to improve the performances of the CNN from its original pre-trained model, a fine-tuning phase is necessary. Fine-tuning takes an already learned model, adapts the architecture and resumes training from the already learned model weights on a different dataset.
First of all we need to download the *train_val* and *solver* prototxt files provided by the author of the same pre-trained model, in our case the VGG16. These files contain information about the architecture with setup parameters useful for the finetuning process, i.e. learning rate multiplier, dropout probability, momentum, etc. By modifying the aforementioned configuration file, we replace the last layer of the CNN by a randomly initialized fully-connected layer with the correct number of face labels to recognize. Moreover, we set the learning rate of the fully connected layer as ten times the learning rate of the rest of the CNN and we set the global learning rate to one tenth of the original one.
From the *train_val* prototxt we notice that the input dataset is in Lightning Memory-mapped Database (LMDB) format file. For this reason, a function to convert our images into a LMDB file is necessary.
After setting up the solver and the caffe prototxt, the model needs to be trained for few epochs, until the convergence of the losses is reached. The final fine-tune command is:

```
caffe train --solver=$SOLVER --weights=$CAFFEMODEL
```

Once fine-tuned, the new model is used to extract the features of the *fc7* layer for each input images and feed the Long-Short Term Memory (LSTM) Recurrent Neural Network (RNN).

4.6 Dimensionality Reduction

The fully connected *fc7* layer of the VGG-16 network produces a 4096 dimensional vector. This vector is input of the LSTM. One may claim that a 4096 feature vector is too big to be efficiently treated by a LSTM. To tackle this point, some extension to reduce the feature dimensionality are proposed.

PCA

Principal Component Analysis (PCA) is a multivariate statistical procedure that is often useful in identifying patterns in high-dimensional data or in reducing dimensionality. In this second case, the new coordinate axes (along which the data varies the most) are called *principal components* and are, by construction, orthogonal. PCA can be usefully used in my case to convert the *fc7* feature vector in a lower dimensional vector and save space and computational time. Figure 9 shows a sketch of the aforementioned system, where the 4096 dimensional output of the CNN is reduced (transformed into a low

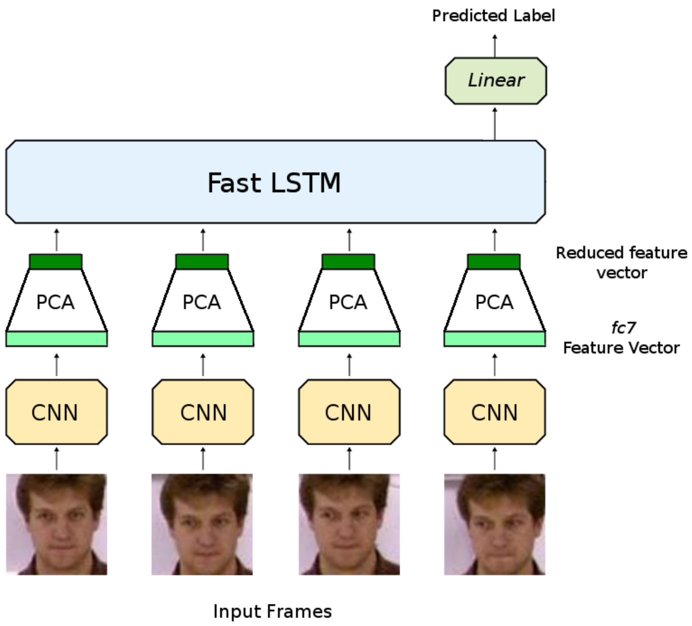


Figure 9: CNN + PCA + LSTM

dimensional vector) and placed as input for the LSTM.
The implementation of PCA is rather straightforward thanks to the python tools provided by the scientific community. In order to implement PCA we may use the *sklearn* python package.
PCA is a powerful tool for reducing the dimensionality. Nevertheless, its linearity may also cause a loss of relevant information in the hyper-dimensional feature vector, leading to a loss of the learnt features and the production of a meaningless vector representation. Moreover there is another intrinsic problem concerning PCA: it does not take into account class information when calculating the principal components. Especially in cases when the differentiating characteristics of the classes are not reflected in the variance of the variables, PCA may not be a good choice of data processing.

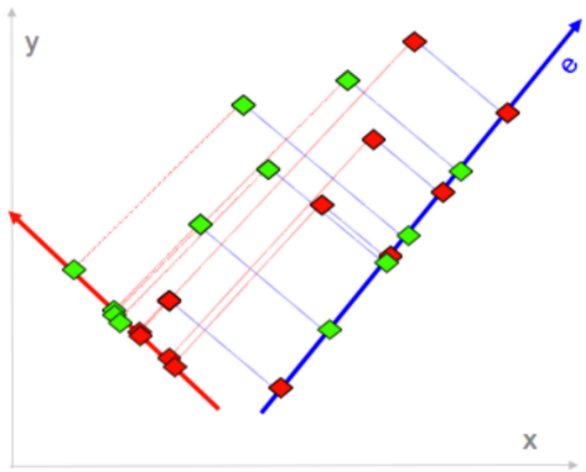


Figure 10: PCA failure example

Figure 10 shows an example of data distribution in 2D, where we have two original dimensions x and y and two classes *red* and *green*. The dimension with the highest variance is the blu axis e, which PCA will pick as the first principal component. However, it is evident that if we use only this principal component, it will make classification more difficult as the data points from red and green classes are dispersed quite evenly along this principal component. The best choice would be the red orthogonal axes, with the lowest data variance.
Given the unclear results of applying PCA, other dimensionality reduction option are proposed.

Artificial Neural Network

Another possible method to perform dimensionality reduction is represented by an Artificial Neural Network placed between the VGG16 and the LSTM. The small ANN module takes the *fc7* feature vector (output of the CNN VGG-16) and produces a new (smaller) feature representation, which would be the new input of the LSTM. It would be trained to classify the video sequences frame-by-frame and, as for the pre-trained CNN, feature vectors would be extracted from the last fully connected layer, where the feature abstraction is higher. The design of such architecture, the setting of the number of hidden layers and hidden units along with the choice of the network parameters, would be performed at implementation time, according to the performances observed. Figure 11 shows a sketch of this second system.

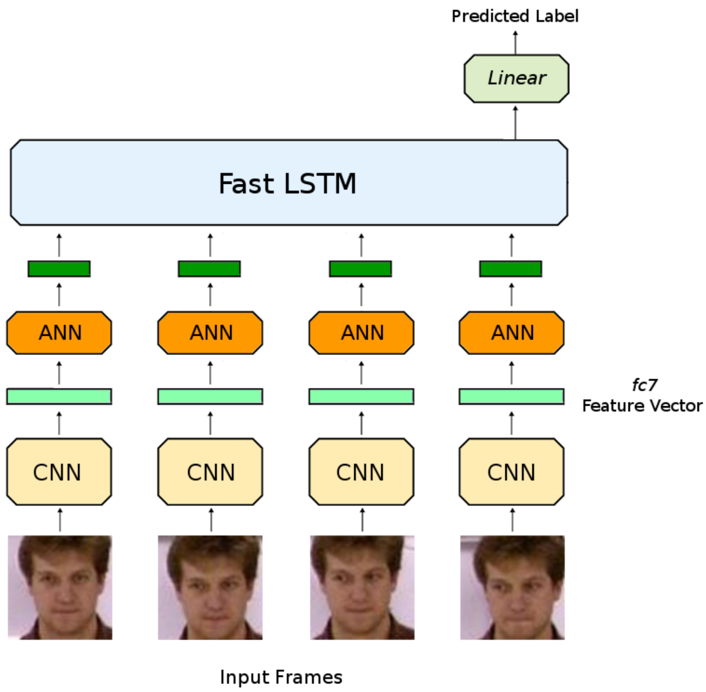


Figure 11: CNN + ANN + LSTM

Convolutional Autoencoders

Autoencoders are artificial neural networks used in unsupervised learning to produce input data representations (encoding), typically for the purpose of dimensionality reduction. An useful version of autoencoders is called Convolutional Autoencoders, which extends the encoding process to two dimensions. Here the standard steps are: input → convolution → deconvolution → error measure. Some implementations perform also pooling, in a different and more complex architecture: input → convolution → pooling → unpooling → deconvolution → error mearure. The implementation of a convolutional autoencoders is straighforward thanks to the python ML libraries (sklearn).

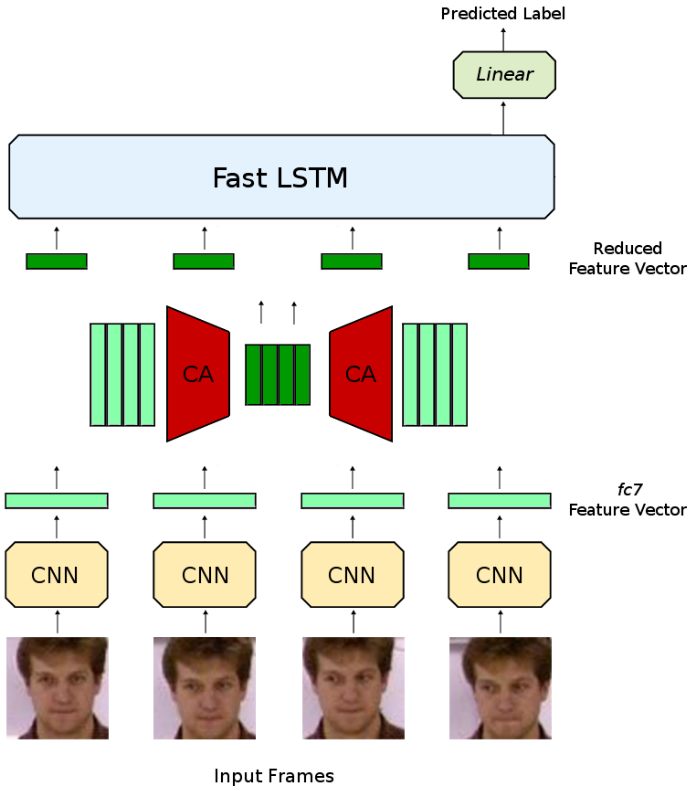


Figure 12: CNN + CA + LSTM

4.7 Work Plan

Figure 13 shows a temporal organization of the work flow that I propose to follow in order to accomplish the specific tasks of which my project is composed. Some of those tasks have been already achieved and are presented in this work.

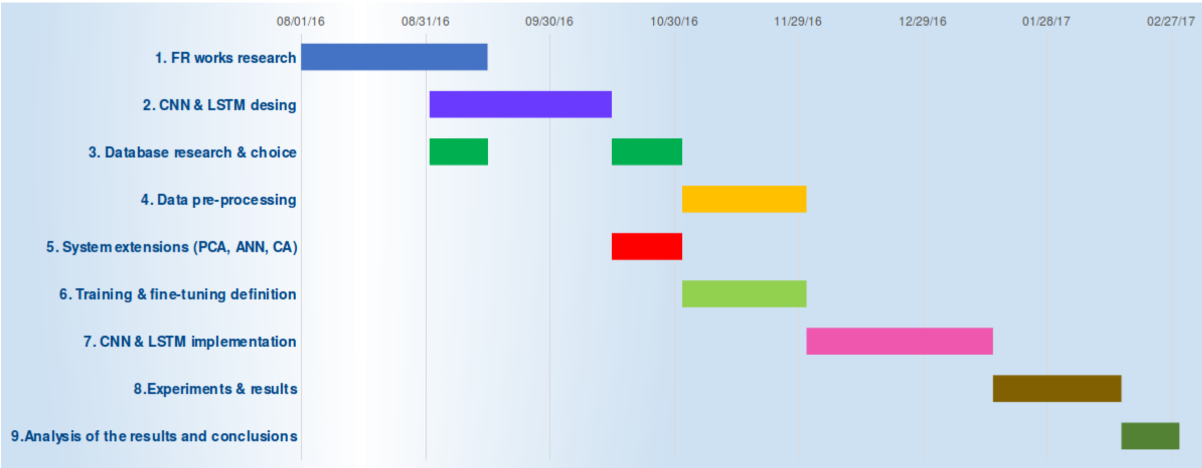


Figure 13: Work plan

5 RESULTADOS

5.1 Aportes y Resultados Esperados

The aim of this work is to investigate the performances of a hybrid Recurrent-based deep method for video face recognition. In addition, we investigate whether a LSTM network can improve the performances of a CNN architecture. Thus, the contributions to the community would be a detailed analysis of the performances of a new hybrid deep temporal framework for video face recognition, along with an exhaustive investigation about how and in which measure temporal information can improve the performances of a CNN model. The experiments would/will be divided into two groups:

1. Test the CNN+LSTM system on the dataset and boost its performances by fine-tuning the hyperparameters.

2. Compare the performances of the CNN alone against the performances of the whole system.

During the fine-tuning step, as introduced at point 1., a grid search among the most important hyperparameter i.e. temporal window, network complexity and dropout, is necessary. In conclusion, the expected results are an improvement of the accuracy of an existing CNN-based model and a substantial improvement of the performances of the CNN by the introduction of the LSTM network.

5.2 Formas de Validación

In order to validate our method and to compare it with other state-of-the-art method, the choice of the evaluation metric and validation strategy is important. Table 6 show the most widely-used evaluation metrics for face recognition.

Table 6: Evaluation Metrics

Metric	**Definition**	**Usage**
Error Rate	$\frac{\text{\# of misclassifications}}{\text{\# samples in val set}}$	General accuracy evaluation
F1 Score	$\frac{2\times\text{true positive}}{(2\times\text{true positive})+\text{false negative}+\text{false positive}}$	Used to give a summary of the Precision-Recall (PR) curve.
ROC / PR curve	$\text{Precision} = \frac{\text{true positive}}{\text{true positive}+\text{false positive}}$ $\text{Recall} = \frac{\text{true positive}}{\text{true positive}+\text{false negative}}$	Used to show the overall performances of an algorithm as its discrimination threshold is varied.

In order to evaluate the performances of the system we decided to calculate the Error Rate and the F1 score, which give us respectively an estimation of the accuracy and an overall knowledge of the precision-recall curve. To calculate the F1 score, the computation of the confusion matrix is necessary.
Table 7 shows the different validation methodologies commonly used for model validation and model selection. Those strategy would be used when finetuning the hyperparameter and for the final calculation of the performances.
Our experiments would be performed with a k-fold cross-validation, with k equals 10. It is also possible to perform a customized leave-one-*video*-out cross-validation, or a more specific strategy following the paper to compare with. In case of MoBo dataset, for instance, the training and test set would be composed by splitting the original dataset folllowing the rule: one activity for training and 3 activity for testing.

Table 7: Validations

Validation	Definition and Usage
LpO CV	Leave-p-out cross-validation uses p obsarvation as the validation and the remaining observations as the training set.
LOOCV	Leave-*one*-out cross-validation is a particular case of LpO CV where $p = 1$
k-fold CV	In k-fold cross-validation the original sample is randomly partitioned into k equal sized sub-samples. The validation process is repeated k times, taking $k - 1$ partitions as training and 1 as test
Monte Carlo CV	Repeater random sub-sampling cross validation, aklso known as Monte Carlo cross validation, randomly splits the dataset into training and validation data. Results are averaged over the splits.

6 RECURSOS

6.1 RECURSOS DISPONIBLES

Señale medios y recursos con que cuenta el Departamento de Informática de la UTFSM, para realizar el proyecto de tesis (libros, software, laboratorios, etc.).

6.2 RECURSOS SOLICITADOS

Señale medios y recursos no disponibles en el Departamento de Informática de la UTFSM, necesarios para realizar el proyecto de tesis (libros, software, laboratorios, etc.).
Su extensión no debe exceder el espacio disponible

References

[1] *Face recognition homepage.* http://www.face-rec.org/general-info/.

[2] *Vgg face descriptor - university of oxford.* http://www.robots.ox.ac.uk/ vgg/software/vgg$_f$ace/.

[3] M. W. J. S. Alex Graves, Christoph Mayer and B. Radig, *Facial expression recognition with recurrent neural networks*, International Workshop on Cognition for Technical Systems, (2008).

[4] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, *End-to-end attention-based large vocabulary speech recognition*, CoRR, abs/1508.04395 (2015).

[5] B. Ben Amor, K. Ouji, M. Ardabilian, and L. Chen, *3D Face recognition by ICP-based shape matching*, in The second International Conference on Machine Intelligence (ACIDCA-ICMI'2005), Nov. 2005.

[6] Y. Bengio, P. Simard, and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult*, IEEE transactions on neural networks, 5 (1994), pp. 157–166.

[7] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, *Unconstrained face recognition: Identifying a person of interest from a media collection*, IEEE Transactions on Information Forensics and Security, 9 (2014), pp. 2144–2157.

[8] L. Chao, J. Tao, M. Yang, Y. Li, and Z. Wen, *Long short term memory recurrent neural network based multimodal dimensional emotion recognition*, in Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge, AVEC '15, New York, NY, USA, 2015, ACM, pp. 65–72.

[9] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using RNN encoder-decoder for statistical machine translation*, CoRR, abs/1406.1078 (2014).

[10] D. C. Corra, D. H. P. Salvadeo, R. L. M. Levada, J. H. Saito, N. D. A, E. Moreira, and S. Carlos, *Using lstm network in face classification problems.*

[11] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, *Long-term recurrent convolutional networks for visual recognition and description*, CoRR, abs/1411.4389 (2014).

[12] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, *Recurrent neural networks for emotion recognition in video*, in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI '15, New York, NY, USA, 2015, ACM, pp. 467–474.

[13] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, *Learning precise timing with lstm recurrent networks*, JMLR, 3 (2002), pp. 115–143.

[14] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, *Learning precise timing with lstm recurrent networks*, J. Mach. Learn. Res., 3 (2003), pp. 115–143.

[15] G. Goswami, R. Bhardwaj, R. Singh, and M. Vatsa, *Mdlface: Memorability augmented deep learning for video face recognition*, in Biometrics (IJCB), 2014 IEEE International Joint Conference on, IEEE, 2014, pp. 1–7.

[16] A. Graves, A. Mohamed, and G. E. Hinton, *Speech recognition with deep recurrent neural networks*, CoRR, abs/1303.5778 (2013).

[17] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra, *DRAW: A recurrent neural network for image generation*, CoRR, abs/1502.04623 (2015).

[18] S. Hochreiter, *Untersuchungen zu dynamischen neuronalen netzen*, Diploma, Technische Universität München, (1991), p. 91.

[19] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[20] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, *Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments*, in Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, Oct. 2008, Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

[21] Z. Huang, R. Wang, S. Shan, and X. Chen, *Face recognition on large-scale video in the wild with hybrid euclidean-and-riemannian metric learning*, Pattern Recognition, 48 (2015), pp. 3113–3124.

[22] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, *On using very large target vocabulary for neural machine translation*, CoRR, abs/1412.2007 (2014).

[23] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, R. Chandias Ferrari, M. Mirza, D. Warde-Farley, A. Courville, P. Vincent, R. Memisevic, C. Pal, and Y. Bengio, *Emonets: Multimodal deep learning approaches for emotion recognition in video*, Journal on Multimodal User Interfaces, 10 (2016), pp. 99–111.

[24] V. Kazemi and J. Sullivan, *One millisecond face alignment with an ensemble of regression trees*, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, Washington, DC, USA, 2014, IEEE Computer Society, pp. 1867–1874.

[25] P. Khorrami, T. L. Paine, K. Brady, C. Dagli, and T. S. Huang, *How deep neural networks can improve emotion recognition on video data*, in 2016 IEEE International Conference on Image Processing (ICIP), Sept 2016, pp. 619–623.

[26] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, *Hierarchical committee of deep convolutional neural networks for robust facial expression recognition*, Journal on Multimodal User Interfaces, 10 (2016), pp. 173–189.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, vol. 86, Nov 1998, pp. 2278–2324.

[28] H. Li and G. Hua, *Hierarchical-pep model for real-world face recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR '15, 2015, pp. 4055–4064.

[29] ——, *Hierarchical-pep model for real-world face recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4055–4064.

[30] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, *Eigen-pep for video face recognition*, in Asian Conference on Computer Vision, Springer, 2014, pp. 17–33.

[31] D. Y. Li Deng, *Deep learning: Methods and applications*, tech. rep., May 2014.

[32] R. Min, J. Choi, G. Medioni, and J. L. Dugelay, *Real-time 3d face identification from a depth camera*, in Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Nov 2012, pp. 1739–1742.

[33] E. G. Ortiz, A. Wright, and M. Shah, *Face recognition in movie trailers via mean sequence sparse representation-based classification*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3531–3538.

[34] S. Otte, M. Liwicki, and A. Zell, *Dynamic Cortex Memory: Enhancing Recurrent Neural Networks for Gradient-Based Sequence Learning*, Springer International Publishing, Cham, 2014, pp. 1–8.

[35] O. M. Parkhi, A. Vedaldi, and A. Zisserman, *Deep face recognition*, in British Machine Vision Conference, vol. 1, 2015, p. 6.

[36] H. Ranganathan, S. Chakraborty, and S. Panchanathan, *Multimodal emotion recognition using deep learning architectures*, Institute of Electrical and Electronics Engineers Inc., United States, 5 2016.

[37] M. Ren, R. Kiros, and R. S. Zemel, *Exploring models and data for image question answering*, in Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 2953–2961.

[38] H. Sak, A. W. Senior, and F. Beaufays, *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition*, CoRR, abs/1402.1128 (2014).

[39] A. Seal, S. Ganguly, D. Bhattacharjee, M. Nasipuri, and D. K. Basu, *A comparative study of human thermal face recognition based on haar wavelet transform (HWT) and local binary pattern (LBP)*, CoRR, abs/1309.1009 (2013).

[40] L. Shang, Z. Lu, and H. Li, *Neural responding machine for short-text conversation*, CoRR, abs/1503.02364 (2015).

[41] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR, abs/1409.1556 (2014).

[42] 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, 2014, IEEE Computer Society, pp. 1891–1898.

[43] ——, *Deeply learned face representations are sparse, selective, and robust*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2892–2900.

[44] Y. Sun, X. Wang, and X. Tang, *Sparsifying neural network connections for face recognition*, CoRR, abs/1512.01891 (2015).

[45] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, CoRR, abs/1409.3215 (2014).

[46] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, *Deepface: Closing the gap to human-level performance in face verification*, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[47] L. Theis and M. Bethge, *Generative image modeling using spatial lstms*, in Advances in Neural Information Processing Systems 28, Jun 2015.

[48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, *Show and tell: A neural image caption generator*, CoRR, abs/1411.4555 (2014).

[49] J. Wang, L.-C. Yu, K. R. Lai, and X. jie Zhang, *Dimensional sentiment analysis using a regional cnn-lstm model*, in ACL, 2016.

[50] L. B. Wolff, D. A. Socolinsky, and C. K. Eveland, *Face Recognition in the Thermal Infrared*, Springer London, London, 2005, pp. 167–191.

[51] C. Xu, S. Li, T. Tan, and L. Quan, *Automatic 3d face recognition from depth and intensity gabor features*, Pattern Recognition, 42 (2009), pp. 1895 – 1905.

[52] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua, *Neural aggregation network for video face recognition*, arXiv preprint arXiv:1603.05474, (2016).

[53] ——, *Neural aggregation network for video face recognition*, arXiv preprint arXiv:1603.05474, (2016).

[54] S. H. Yoon, G. T. Hur, and J. H. Kim, *Recurrent Neural Network Verifier for Face Detection and Tracking*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 488–499.

[55] Z. Yu and C. Zhang, *Image based static facial expression recognition with multiple deep network learning*, in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI '15, New York, NY, USA, 2015, ACM, pp. 435–442.

[56] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, *A C-LSTM neural network for text classification*, CoRR, abs/1511.08630 (2015).