Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 18, Iss. 1, pp. 47-59, 2020
© Sociedade Brasileira de Redes Neurais

Soft Biometrics Classification in Videos Using Transfer Learning and Bidirectional Long Short-Term Memory Networks

Marcelo Romero*, Matheus Gutoski*, Leandro Takeshi Hattori*, Manassés Ribeiro, Heitor Silvério Lopes***
*Federal University of Technology Paraná, Curitiba, Brazil
**Catarinense Federal Institute of Education, Science and Technology, Videira, Brazil
nmarceloromero@gmail.com, matheusgutoski@gmail.com, leandrotakeshihattori@gmail.com
manasses@ifc-videira.edu.br, hslopes@utfpr.edu.br

Abstract – Soft biometrics attributes can be useful to perform identification of individuals, since they provide information that can be used to differentiate one individual from another without intrusiveness. Moreover, the large number of surveillance cameras installed in public places allows to acquire videos in real time without much effort. However, this demands an exhaustive process of analysis to be carried out by one or more human observers, which makes necessary the use of methods capable of performing the task automatically. Deep Learning methods have risen in the recent years, achieving state-of-the-art performances for several computer vision tasks such as object recognition, object detection, and image segmentation. This work aims at empirically studying the suitability of a DL approach to perform soft biometrics classification in videos. We evaluate the use of a DL model to learn temporal dependencies, in order to perform soft biometrics classification in videos. For this purpose, we present an approach based on the use of a pre-trained Convolutional Neural Network as feature extractor in combination with a Bidirectional Long Short-Term Memory network to perform the classification.

Keywords – Deep Learning, Convolutional Neural Network, Bidirectional Long Short-Term Memory, Transfer Learning, Soft Biometrics

1 Introduction

In recent years, the necessity to increase public security led to the growth of the number of surveillance cameras installed in public places [1]. They allow obtaining images and videos in real time without much effort. Hence, information can be extracted from these resources to solve different types of problems, such as the identification of individuals in videos. For this task, soft biometrics are valuable, since they provide useful information for identifying individuals. Soft biometrics are human characteristics, physiological or behavioural, that are distributed in classifiable human-understandable pre-defined types [2]. Although soft biometrics are usually not unique for each individual, they can provide prior information about the subjects that can lead to their identification. Notwithstanding, using just one soft biometric may not be a suitable option to identify a particular individual, but when used in combination they can lead to achieve relevant results to improve a recognition task under highly variable conditions. Soft biometrics can also be used to complement other primary biometric identifiers such as fingerprints [3] and, on the opposite to primary biometrics, they are usually non-intrusive in the sense that they do not require direct participation of the individual to be gathered. This can be accomplished, for instance, through the use of surveillance cameras. The classification of soft biometrics is a relevant research topic since it provides an alternative to the manual analysis of surveillance videos, which is an exhaustive process to be carried by a human observer. Therefore, computer vision plays an important role to tackle this problem.

Computer vision methods provide means to extract and process low-level features from images or videos in order to produce high-level semantic information [4]. Traditional image or video classification is based on a series of steps that include data pre-processing, feature extraction, and classification [5]. In addition to the traditional pattern recognition methods, Deep Learning (DL) approaches [6] have risen during the past years achieving the state-of-the art performance for several computer vision tasks such as object detection and recognition [7–9], and image classification [10–12]. These methods differ from the traditional ones in their learning process, since they are end-to-end classifiers. This implies that they can learn both, the features and the classifier, during the training of the model [6].

The objective of this work is to use a combination of DL models to perform soft biometrics classification in videos. For this purpose, we propose an approach based on a pre-trained Convolutional Neural Network as the feature extractor combined with a Bidirectional Long Short-Term Memory network as the classifier. We aim at exploiting the capability of CNNs to represent high-level features and the capability of BLSTMs to learn temporal dependencies for the specific problem of classifying soft biometrics in videos. Our approach could aid at ultimately performing the tracking of individuals by only considering attributes such as clothes length or colour. Hence, we also apply object detection in order to classify, at each frame, only the soft biometrics of an individual within a bounding box, obtained through an object detection model. This work also introduces a video dataset, which is publicly available to foster future research. Considering these issues, the contributions of this work are: (1) to apply transfer learning in order to extract features to perform soft biometrics classification, (2) to introduce a method based on using pre-trained CNN and BLSTM networks to perform soft biometrics classification in videos, (3) to serve as an initial step towards performing automatic tracking of individuals considering their soft biometrics.

This work is structured as follows: a brief introduction of different aspects related to the methods applied in this work are presented in Section 2. The related literature is presented in Section 3. The methodology is described in Section 4. The experiments are presented in Section 5 along with the obtained results. Finally, general conclusions regarding the work and proposals for future work are presented in Section 6.

2 Background

This section describes relevant aspects regarding the methods used in this work. First, we briefly describe Convolutional Neural Networks (CNNs). Then, a specific CNN architecture named Inception-v3 is presented. After, we introduce the concepts behind the Single Shot MultiBox Detector (SSD), which is a neural network developed to perform object detection in images. Finally, we introduce Recurrent Neural Networks (RNNs) and two widely known types of RNNs: Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BLSTM) networks.

2.1 Convolutional Neural Networks

CNNs [13] are feed forward neural networks that allow to deal with input data of n dimensions, such as images or videos. CNNs learn hierarchies of features, obtaining more abstract representations of the original data. This allows to feed the model with raw data without the need of previously extracting their features. CNNs are end-to-end classifiers: the features and the classifier are jointly learnt. This allows to obtain a feature extractor that is built specifically based on the data that was used to train the model. A CNN contains Convolutional, Pooling and Fully Connected layers. The first two are present in the first part of the network and they act together as the feature extractor of the model, whilst the Fully Connected layers are used for performing classification. Convolutional layers are composed of a set of learnable filters, also known as kernels, that slide across their input during the forward pass of the network and calculate dot products between their entries and the input at any position. The input of this layer could be 2-dimensional or 3-dimensional arrays, such as grey-scale or colour images. The output is composed of a set of concatenated activation maps, which are produced by each filter. Since convolutions are linear transformations, a non-linearity is applied to their output. The most used activation function is ReLU. On the other hand, Pooling layers are inserted between convolutional layers to progressively downsample the data representation in order to reduce the amount of parameters of the model. This is done to reduce the computation required to train the network and to control overfitting, which occurs when the model is over-trained and loses its generalisation capability [14]. The Fully Connected layers, also known as dense layers, apply the dot product between the output of the previous layer and the weights connected to its processing units, a bias term added to the result of the product operation and finally a non-linear activation function is applied. The first dense layer receives as input the output of the last layer of the feature extractor part of a CNN. Often this value is a set of feature maps, which are previously flattened before being fed to the first dense layer. The final layer of a CNN is also a Dense layer. For classification tasks, the non-linearity of the last layer is usually the Softmax function. The function allows to squash a K-dimensional input y to a K-dimensional vector $\sigma(y)$ containing real values in the range $[0..1]$ that add up to 1. The label of the class with the highest probability is employed as the final classification.

2.2 Inception-v3

Inception-v3 [15] is a CNN architecture with Convolutional and Pooling layers in the first part of its structure. Local Response Normalization (LRN) [16] operations are also used in this first part of the model. The next part of the network contains sequentially stacked Inception modules introduced in [17]. Inception modules are layers that apply Convolutions and Pooling operations on the same input, and the outputs of each of them are concatenated to be fed to the next layer. Occasional Max-Pooling layers with stride two are also used between the Inception modules. With this structure, the network decides itself which operation is more suitable for the task during the training phase. It also allows learning both local features and more abstract ones through small and large convolutions, respectively. In this work, the output of the last Inception module is used as a feature vector describing a raw RGB input image of size $299 \times 299 \times 3$.

Inception-v3 is a very deep network: it is composed of 42 layers. To avoid the vanishing gradient problem, caused by the depth of the network, the original Inception includes auxiliary classifiers connected to its intermediate layers [17]. These classifiers are mini CNNs that are connected to two Inception modules of the network. This strategy incites the network to strengthen the classification capability of its lower layers. During the training phase, the loss of the classifiers are weighted and added to the total loss of the network. For the inference, the auxiliary classifiers are discarded. The Inception-v3 architecture also include these auxiliary classifiers. However, they do not seem to improve the convergence early in the training and do not help evolving low-level features as it was initially assumed in [17]. Instead, presumably they act as regulariser. The complete architecture of Inception-v3 is described with detail in [15].

2.3 Single Shot MultiBox Detector

SSD[1] is a feed-forward CNN used for object detection within images. The network is capable of detecting 20 different classes of objects in complex backgrounds. However, only the detection of one specific class, which is that of individuals (humans), is used in this work. The structure of SDD is composed of two main parts: the base network and the detector. The base network

[1] https://github.com/weiliu89/caffe/tree/ssd

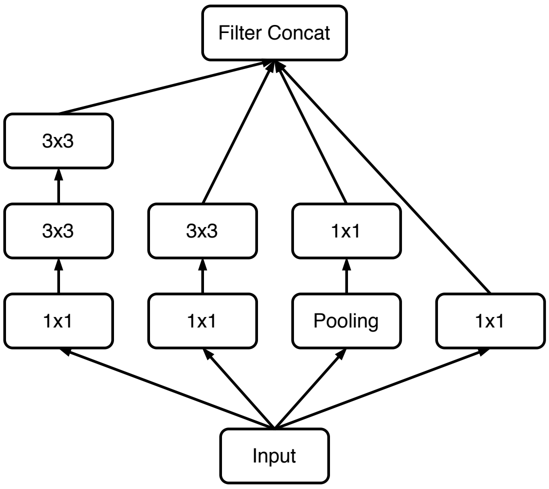

Figure 1: An Inception module used in the Inception-v3 architecture.

is a CNN designed for image classification, more specifically the VGG16 network [18]. The detector is formed by an auxiliary structure that uses multi-scale feature maps, convolutional predictors and default boxes associated with feature map cells [8]. Figure 2 presents an image with the objects detected by a detection algorithm.

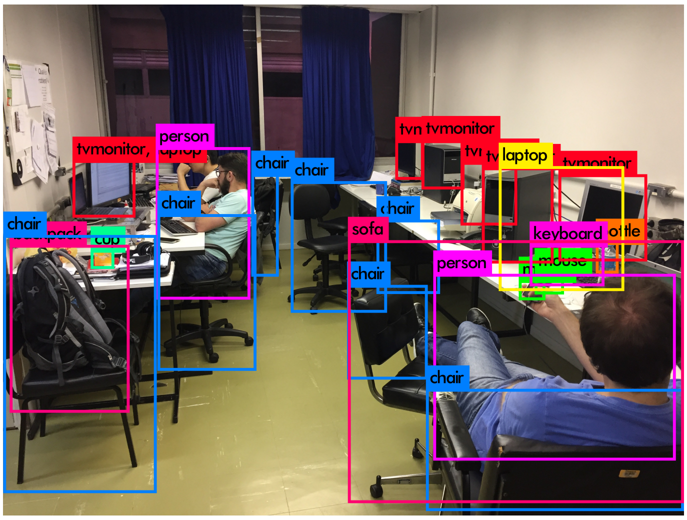

Figure 2: Sample image with detected objects by the SSD network. Several classes can be identified such as Chair, TV Monitor, Laptop, and Sofa. In this work, we focus on the Person class.

2.4 Recurrent Neural Networks

RNNs are a type of neural network with cyclic connections between their processing units [19]. These recurrent connections allow to store information related to past inputs for an amount of time that is not initially fixed. Instead, it depends on the weights of the network and the input data. The structure of RNNs allows to operate over sequences of input and output data, unlike feed forward neural networks, which can only receive fixed-sized vectors as input and produce fixed-sized vectors as outputs. A standard RNN receive as input a vector $\vec{x}$ and produces an output vector $\vec{y}$, which is influenced not only by the current input but, also, by the history of inputs that were previously fed to the model. This is accomplished by a hidden state vector $\vec{h}_t$ in a given time t. Figure 3 shows a simple RNN architecture.

Given the input $\vec{x}_t$ at a time t, the hidden state $\vec{h}_t$ of the RNN is defined by Equation 1, where $\vec{W}_{hx}$ and $\vec{W}_{hh}$ are the weights connected to the input of the network and the previous hidden state respectively, and $\vec{h}_{t-1}$ is the hidden state at the time $t-1$.

$$\vec{h}_t = tanh(\vec{W}_{hh}\vec{h}_{t-1} + \vec{W}_{hx}\vec{x}_t), \tag{1}$$

2.5 Long Short-Term Memory Networks

LSTMs are capable of learning long-term dependencies through a more complex structure based on special hidden units known as memory cells [20]. The internal structure of memory cells permits to overcome the vanishing (or exploding) gradient problem [21], allowing the network to retain dependencies from early inputs. A LSTM memory cell is shown in Figure 4.

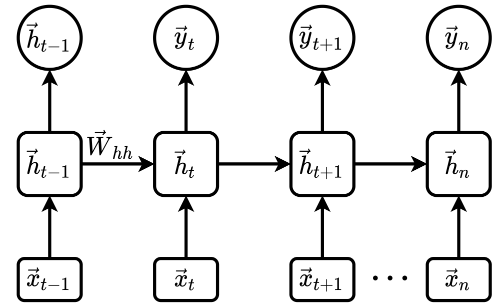

Figure 3: Architecture of a Recurrent Neural Network.

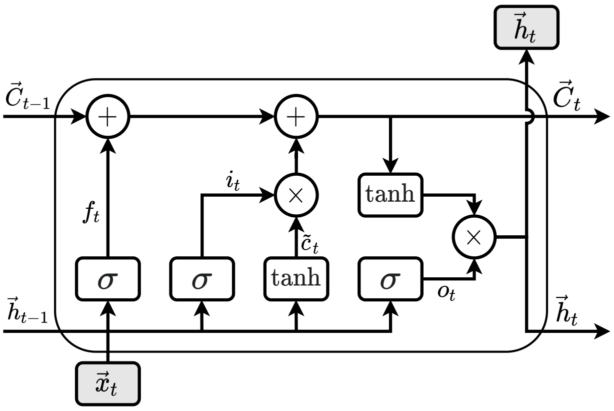

Figure 4: A LSTM memory cell.

LSTM cells allow to optionally remove or save information, which is accomplished through a mechanism based on gates. A cell is composed of the following gates: forget gate (f_t) to forget information, input gate (i_t) also known as update gate, output gate (o_t), and a gate that generates the internal cell state ($\tilde{c}_t$). The weights matrices ($\vec{W}_f$, $\vec{W}_i$, $\vec{W}_c$, $\vec{W}_o$) and bias terms ($\vec{b}_f$, $\vec{b}_i$, $\vec{b}_c$, $\vec{b}_o$) are also used, each one associated to a certain gate. The Sigmoid (σ) and Hyperbolic Tangent ($tanh$) are the activation functions used in the cell.

2.6 Bidirectional Long Short-Term Memory Networks (BLSTM)

Although LSTM networks are a widely used DL method due to their capability to learn semantics with long-term dependencies, they have a weakness: they consider only previous information without exploring future contexts [22]. Thus, for an input at the time t belonging to a sequence of length n, the LSTM will consider only the inputs from times $t-1, t-2, ..., t-n$. If there is available information from the future, for instance inputs at the times $t+1, t+2, ..., t+n$, the network will not consider them to produce the output. To tackle this issue, BLSTM were designed to take advantage of both previous and future information by using a structure based on two separate hidden layers in order to process the data. Figure 5 presents a simple BLSTM. Basically, it is a recurrent network with two LSTM cells, one performs a forward propagation whilst the other propagates the data backwards. The output of each cell is concatenated in order to produce the final output of the layer. More backward-forward layers can be stacked to form a deep BLSTM, with the output of each layer used as input for the next one.

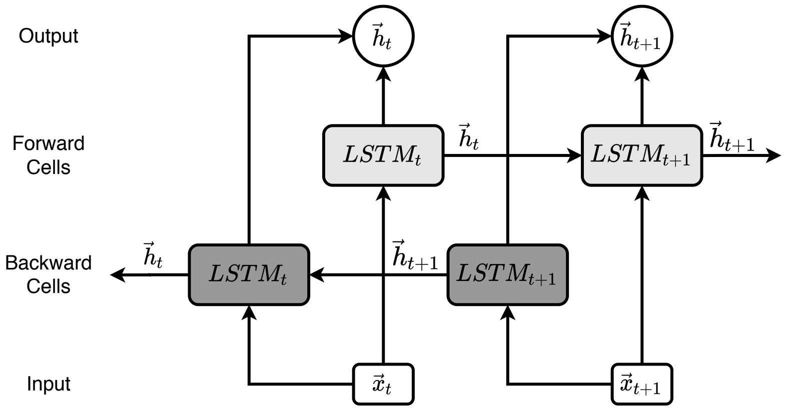

Figure 5: Bidirectional RNN composed of LSTM cells.

3 Related Work

Transfer learning is a paradigm that employs knowledge from previous experiences for solving new problems. Traditional machine learning algorithms assume that the train and test sets lie within the same feature space distribution [23], whilst transfer learning considers that the distributions may be different [24, 25]. Despite not belonging to the same distribution of the original train data, some tasks can be solved by using previously acquired knowledge, given that the new problem is somewhat similar to the original one. One of the advantages of this approach is that it relieves the need for annotated data, which is usually costly in real-world applications.

To the best of our knowledge, no work has been carried out with the objective to address the use of transfer learning to perform soft biometrics classification in videos. Although DL approaches based on training a model from scratch to perform this task can be found in the recent literature. For instance, the work by [26] presented two CNNs architectures for soft biometrics classification: one for simultaneously classifying three soft biometrics (Upper Clothes, Lower Clothes, and Gender) and the other for classifying a single soft biometric. The work presented by [27] estimated the age and gender of individuals from images of human faces, also using a CNN. Both works present very traditional CNN architectures composed of convolutional and pooling layers that inputs raw images and propagates the data through the network until reaching the classifier part, composed of dense layers. On the other hand, works by [28] and [29] presented more complex architectures, which are based on dividing an image into patches and training the feature extractor part of the CNN (convolution and pooling layers) separately for each patch and joining the features yielded by the last pooling layer into a unique vector which, in turn, is fed to the classifier part of the network.

Since DL methods drastically improved the performance of image classification frameworks, the same occurred with video classification, since a video is just a sequence of images. Considering this line, several works published in recent years present approaches for performing this task using Recurrent networks such as LSTM, often in combination with CNNs, since they allow to learn temporal dependencies. For instance, [30] introduced a hybrid framework able to learn static spatial information, short-term motion and long-term temporal dependencies in videos. The approach combines two CNNs, the first for learning spatial features and the second for learning motion features, they to obtain a regularised fusion network that is connected to LSTM networks, allowing to learn the temporal clues. In [31], two methods for performing video classification were proposed. The first is based on CNN architectures that incorporate temporal feature pooling, which consists in applying max-pooling operations across a set of frames from a video clip. The AlexNet [16] and GoogLeNet [17] architectures were used to process individual frames in the work. The second approach introduced in the paper is based on the use of five stacked LSTM layers that receive the output of the final layer of a CNN at each consecutive frame from a video. At each time step, a softmax layer produces the class prediction. Since the LSTM-based architecture produces a prediction per frame, for the inference of the whole video clip the authors evaluated several approaches to combine individual frame predictions, such as returning the prediction at the final time step, and applying a max-pooling operation on the predictions over time, among others. The best strategy was achieved by weighting the predictions over time by a threshold, then summing and returning the max. The work by [32] also takes advantage of LSTMs for visual recognition and description. The authors presented a Long-term Recurrent Convolutional Network, which basically combines both CNNs and LSTMs in a single network that is trained end-to-end, in order to perform image description, activity recognition or video description. In all cases, the network does not produce a classification score. Instead, it uses the capabilities of the LSTM part of the model to generate a sequence of words describing the image or video.

Other works addressing the querying of soft biometrics in videos or images were proposed during the recent years. The work by [33] proposed a person retrieval method. The method is based on the use of Mask R-CNN [34] to detect and segment persons in images from surveillance videos. The height and torso colour are used as parameters to filter individuals in the images, and they are calculated using the Tsai camera calibration approach [35] and a fine-tuned AlexNet [36] for colour classification, respectively. A final filter based on classifying the gender of the person is applied is necessary, which is also classified using AlexNet. On the other hand, the work by [37] focused on the use of the DenseNet network [38] to classify a set of images containing soft biometrics defined by a textual query. The network was trained using ImageNet dataset [39] and an additional Fully Connected layer with 512 neurons has been used to serve as classifier. The input of the model is a set of attributes to serve as query for the extraction of images of individuals containing the selected soft biometrics. The approach is tackled following a multi-task approach, thus the network is intended to perform multi-class classification.

Similarly to the works presented in this Section that are based on combining a CNN with a LSTM, this work also pursuits to take advantage of the capability of CNNs to perform image-related tasks and the capability of LSTM networks to learn temporal dependencies. Notwithstanding, our approach is not based on training an end-to-end classifier, instead, we use a pre-trained CNN to work as feature extractor. The features are used as input for a BLSTM network, a type of LSTM that is able to take advantage of both previous and future information, which is trained so as to be able to classify soft biometrics from sequences of feature vectors. This work also applies a previous step: individual detection. Before applying feature extraction or some classification technique on a frame (or a sequence of frames), we apply an object detection algorithm, based on the use of a pre-trained CNN specifically devised for such task, in order to isolate a bounding box containing the individual for whom the soft biometric attribute is going to be classified. Some issues are derived from its application, which are going to be detailed in the following sections. It is worth mentioning that we intend this to be an initial step to foster future works to combine soft biometrics classification with individual detection in order to perform individual tracking by only considering some visual attribute.

4 Methodology

In this work, two pre-trained CNNs were used to perform soft biometrics classification in videos: SSD and Inception-v3, for individual detection and feature extraction purposes respectively. A BLSTM is used as the classifier. The BLSTM inputs a sequence of feature vectors extracted using the pre-trained CNN applied on bounding boxes obtained with SSD, and outputs a classification result for the sequence. This section describes this process in more detail.

4.1 Individual Detection

We apply SSD on each frame of a video to detect all individuals within the frame. If more than one individual is detected, we take the one that is within the biggest bounding box, which, ideally, is the individual that is closest to the camera. Although selecting only the biggest bounding box allows obtaining most of the boxes that correspond to the individual that the video was labelled for, it does not provide a solution in case that individual does not appear among the detected boxes in the frame. In such cases the detector returns a *noise*: a bounding box containing an individual or some other object in the background that the video was not labelled for. Figure 6 presents an example of this situation: the proper individual is detected in frame 4, for which the video was labelled (man within the red bounding box) whilst it is not detected in frame 15, in which case the algorithm will return the biggest bounding box (the one containing the woman in the background). The *noise* problem is tackled in this work through the use of a BLSTM approach, which is explained in Section 4.2.

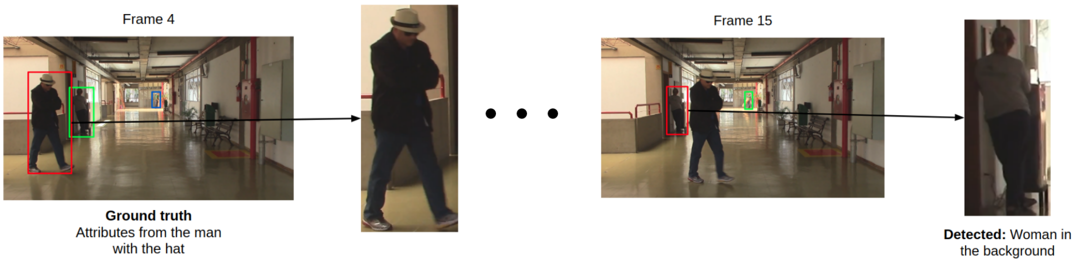
Frame 4
Ground truth
Attributes from the man with the hat
Frame 15
Detected: Woman in the background

Figure 6: Detection using the SSD network. The video has the label for the attributes of the man with the hat (this is the ground truth). In Frame 4, the model detects two individuals, and the selected individual is ground truth. However, the model does not detect the ground truth individual that the video was labelled for in some frames (e.g. Frame 15).

4.2 Transfer Learning with BLSTM Classification

We propose the use of a pre-trained CNN (Inception-v3) as feature extractor and a BLSTM as the classifier. The CNN receives as input a bounding box containing the detected individual within a frame (obtained with SSD), which is previously resized to $299 \times 299 \times 3$. The network then yields a 2048-dimensional feature vector. Note that the CNN is pre-trained with the ImageNet dataset [39]. The vectors are grouped into sets of consecutive sequences and are used as input for the BLSTM. The architecture of the BLSTM produces a score for each sequence of features and the final classification for the entire video is obtained by averaging the scores of all sequences. It is worth mentioning that the *noises* are not removed neither for the training of the BLSTM nor for its further inference. The idea behind this strategy is to improve the capability of the model to treat those *noises*.

Figure 7 presents our approach applied to a sequence k of n frames. The process starts with the extraction of the individual from the frame using SSD (recall that the detection is kept even if it is a *noise*, i.e., the detection is of something other than the main individual on the scene). Each bounding box extracted is then fed to the feature extractor part of the pre-trained Inception-v3, obtaining n feature vectors for a sequence k. The n vectors are then fed to the BLSTM, which produces the score y_k for that sequence. The final classification y for N sequences is calculated through $\frac{1}{N}\sum_{k=0}^{N} y_k$. This process remains unchanged for both training and inference. However, for the training phase, the weights of the BLSTM are updated using a gradient-based optimiser and the error is calculated per sequence instead of per video.

The BLSTM architecture receives as input a feature vector at each time step. Thus, the network inputs a sequence of feature vectors. The input is fed to forward and backward LSTM blocks. The outputs of both blocks are concatenated and connected to the output layer, which produces a 2-dimensional output filtered by a Softmax function, which has a squashing effect over its inputs, converting them into real values ranging from 0 to 1. Each position of the output vector of the function represents the probability that the input of the network belongs to a certain class.

The weights of the BLSTM are initialised through the Glorot method (also known as Xavier) [40]. In order to aid generalisation, we use L2 Regularisation. Dropout (50%) is also applied to the inputs and outputs of the cells during the training phase. The BLSTM was trained using Cross Entropy Loss (CEL) as cost function, presented in Equation 2, where $\vec{y}$ is the ground truth label, and $\hat{\vec{y}}$ is the output of the network for a pattern i considering n patterns fed to the network.

$$CEL = -\frac{1}{n}\left(\sum_{i=1}^{n} \vec{y}_i \cdot \log(\hat{\vec{y}}_i)\right), \tag{2}$$

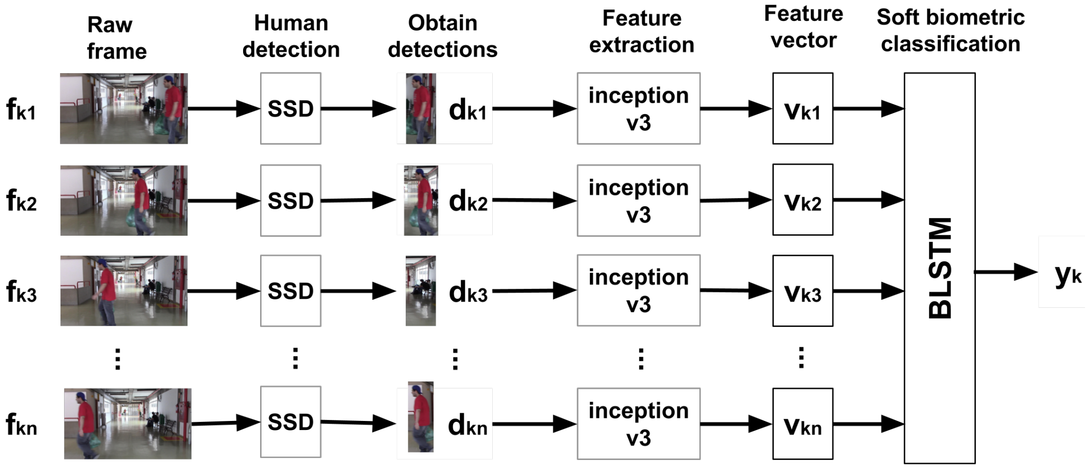

Figure 7: Classification process for the sequence k of n frames. SSD is applied to each frame and the detected individual is passed to Inception-v3 to generate a feature vector that is used as input to the BLSTM model. The output y_k is the score for the sequence. The final score is obtained by averaging the scores of all sequences of the video.

The parameters of the network were optimised using the Adam [41] algorithm, with a learning rate equal to 0.001. Each training process was carried through 100 epochs using a batch size equal to 512.

4.3 Evaluation Metric

Since the approach presented in this work performs binary classification, the final classification result can belong to either positive or negative class. Hence, we use the product of Sensitivity (S_e) and Specificity (S_p), defined in Equations 3 and 4 respectively, as performance metric.

$$S_e = \frac{TP}{TP + FN} \tag{3}$$

$$S_p = \frac{TN}{TN + FP} \tag{4}$$

where True Positives (TP) is the number of positive instances that were correctly classified as such, True Negatives (TN) is the number of negative instances that were correctly classified as such, False Positives (FP) is the number of negative instances that were incorrectly classified as positive and False Negatives (FN) is the number of positive instances that were incorrectly classified as negative.

5 Experiments and Results

This section presents the results obtained by our DL approach proposed for soft biometrics classification. Stratified cross-validation with 10 folds was used to divide the dataset into train and test sets in order to validate the methods. This validation method is a variation of the traditional K-fold cross-validation that aims at preserving the percentage of samples of each class when dividing the dataset. The sequence lengths used for the BLSTM range from 2 to 5 and the following number of processing units for the LSTM cells were tested: 32, 64, 128, 256, 512, 1024 and 2048. Considering this, the possible hyper-parameter combinations add up to 28. Moreover, since the cross-validation process is based on the use of 10 folds, the overall number of experiments carried in this work is equal to 280.

5.1 Implementation

The Python library TensorFlow [42], version 1.3.0, was used to apply the pre-trained Inception-v3 network as feature extractor. TensorFlow was also used to apply individual detection within the videos using the SSD detector. The following Python libraries were also used to evaluate the performance of the models and for image processing: OpenCV2 version 2.4.9, SciPy version 0.19.1, Scikit Image version 0.12.3, and Scikit Learn version 0.18.2.

5.2 Dataset

The dataset used in this work is named UTFPR-SBD2. It contains people walking in a hall of the Federal University of Technology - Paraná. The dataset is available for research purposes in a public repository[2]. This dataset contains 360 videos of

[2] http://labic.utfpr.edu.br/datasets/UTFPR-SBD.html

48 different individuals. The videos have a resolution of 1920×1080 with a frame-rate of 30 FPS. The total number of frames varies for each video, with an average of 66 frames per video. The videos contain individuals walking in four different directions: left-to-right, right-to-left, back-to-front and front-to-back. Figure 8 presents sample frames from this dataset considering the four walking directions.

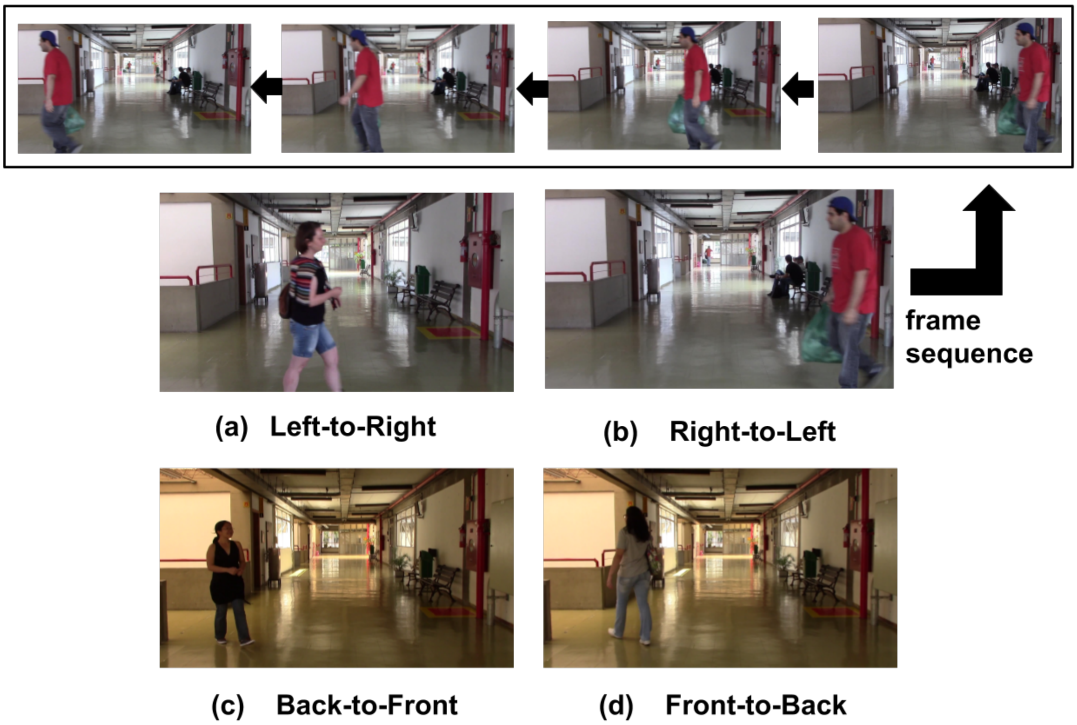

Figure 8: Sample frames from the UTFPR-SBD2 dataset. Each frame corresponds to one of the four walking directions of the individuals.

Each video was manually labelled considering the individual that is walking the closest to the camera. Thus, people walking or sitting in the background are considered as *noise* that must be treated as such for classification tasks. The following labels are provided for the dataset: Gender, Upper Clothes, Lower Clothes, Upper Colour, Lower Colour, Hat, Carrying Object in Hand, Backpack, Glasses, Sunglasses, Long Hair, Hair Colour and Noise (which defines if there are other individuals in the background of the video). All labels represent binary classes, except those referring to colour attributes.

In this work we use the following categories: Gender, Lower Clothes, Upper Clothes, Hat, Long Hair and Carrying Object. These categories were chosen for their popularity and considering that they provide very useful information when tackling the people identification problem in videos. It is worth mentioning that the experiments performed in this work were are intended to tackle only the binary classification problem, thus multi-class labels provided by the datasets were not used. The classes of the attributes Upper Clothes and Lower Clothes may be short (class 0) or long (class 1). As for the Gender attribute, men are represented with the class 1 and women are represented with the class 0. All other labels are Boolean, with class 1 representing the affirmation of that attribute, such as the presence of an accessory or the affirmation that the individual has long hair, and the class 0 representing the contrary.

As for the attributes that reference colours included in the labels of the dataset, there are 10 possible colours represented by an integer value in the range $[0, 9]$. The colours, in sequence (the first is represented with the number 0 and the last one with the value 9), are: red, green, blue, black, white, yellow, pink, colourful (multiple colours), grey and brown. Tables 1 and 2 present the soft biometrics attributes that were annotated for the dataset along with their class distributions:

5.3 Baseline: Discrete Classification

To obtain the baseline results, the Inception-v3 was fine-tuned using a discrete version of the UTFPR-SBD2 dataset, initially obtained by applying SSD to acquire a sequence of bounding boxes containing an individual. All frames that contained *noises* (individuals that the video was not labelled for that can appear in the background) were removed.

Note that we fine-tuned and tested the network using only a frame per video. Hence, four selection strategies were used to select the frame to perform the experiments: first, middle, final and random frame. For all cases, the best results were achieved when using the first frame of each video. These are the results presented in the first column of Table 3.

The fine-tuning process was based on training the final layer of the network using SGD, with learning rate equal to 0.01 and CEL as cost function. Each training process was carried through 200 epochs. These training parameters were defined considering the default values set by the library used to perform the fine-tuning of Inception-v3.

Table 1: Binary labels of UTFPR-SBD2. Numbers in parenthesis represent the number of videos with that label.

Attribute	class 0	class 1
Gender	female (120)	male (240)
Lower clothes	short (52)	long (308)
Upper clothes	short (200)	long (160)
Using hat	no (272)	yes (88)
Backpack	no (281)	yes (79)
Carrying object	no (224)	yes (136)
Using glasses	no (203)	yes (157)
Using sunglasses	no (294)	yes (66)
Have long hair	no (265)	yes (95)
Noise	no (126)	yes (234)

Table 2: Multiclass labels of UTFPR-SBD2. All labels refer to the colour of the label. Numbers in parenthesis represent value used to represent a specific class.

Class	Attribute		
	Upper Clothes	Lower Clothes	Hair
red (0)	61	0	12
green (1)	6	11	0
blue (2)	40	155	0
black (3)	149	154	309
white (4)	60	8	31
yellow (5)	5	0	8
pink (6)	16	0	0
colourful (7)	4	4	0
grey (8)	11	12	0
brown (9)	8	16	0

5.4 BLSTM

This section presents the results obtained by the BLSTM. Table 3 shows the mean $S_e \times S_p$ for all folds of the cross-validation. The Table also includes the standard deviation in order to evaluate the consistency of the classifiers throughout all folds.

Table 3: Mean and standard deviations for the Se×Sp metric for binary classification considering each attribute, achieved by our approach and by the baseline.

Attribute	Baseline	BLSTM
Gender	0.80 ± 0.13	0.98 ± 0.03
Upper Clothes	0.77 ± 0.11	0.97 ± 0.04
Lower Clothes	0.90 ± 0.10	0.99 ± 0.01
Hat	0.65 ± 0.16	0.69 ± 0.08
Long Hair	0.83 ± 0.11	0.96 ± 0.05
Carrying Object	0.73 ± 0.13	0.83 ± 0.12

Although defining the gender of an individual could be a hard task depending on the scene and the quality of the video, our method did not present much difficulty to classify this soft biometric. Results show that even the baseline achieved a good performance (mean $S_e \times S_p$ equal to 0.80). The CNN-BLSTM managed to improve those results by achieving a $S_e \times S_p$ of 0.98, and also reducing the standard deviation.

Considering the attribute *Upper Clothes*, the tendency is the same, with the CNN-BLSTM achieving the best $S_e \times S_p$ results, also achieving the lowest standard deviation.

The *Lower Clothes* is the easiest attribute to classify for the UTFPR-SBD2 dataset: the baseline achieved a very high mean $S_e \times S_p$ equal to 0.90. The BLSTM approach corroborates this insight, achieving a value equal to 0.99. Although the performance improvement when using the BLSTM is minimal, the classification of this attribute does not give too much margin for improvement.

On the opposite to *Lower Clothes*, the *Hat* attribute is the most difficult one to classify. The baseline is the lowest among all soft biometrics attributes. Again, the BLSTM achieved a very low standard deviation. This tendency is repeated for all other soft biometrics attributes considered in this work. The CNN-BLSTM approach provides more accurate results along with a better classification balance, due to the improvement of the $S_e \times S_p$ values when compared to the baseline. The method also tended to show more consistent classification performances throughout the folds of the cross-validation, since it yielded the lowest standard deviation considering all evaluation metrics.

The outstanding results for some attributes could be given by several factors such as the homogeneity of the classes from the dataset in terms of visual appearance and features extracted with Inception-v3 that are very well suited for this specific task. These factors led even the baseline to achieve satisfactory results for some labels, as occurred for the *Lower Clothes* attribute for the UTFPR-SBD2 dataset.

Table 4 presents the classification events, more concretely the True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN) achieved by our approach and by the baseline. In this case, we present the classification result obtained for each sample of the dataset when it was part of the test set during the cross-validation process. Although we already provide values for our evaluation metrics in Table 3, we introduce this information in case that the reader intends to measure the performance of our method through other metrics.

Table 4: TP, FP, FN and TN obtained when each sample was part of the test set during the cross-validation.

Attribute	Baseline				BLSTM			
	TP	FP	FN	TN	TP	FP	FN	TN
Gender	104	16	19	221	118	2	0	240
Upper Clothes	193	7	33	127	197	3	2	158
Lower Clothes	48	4	6	302	52	0	1	307
Hat	220	52	17	71	265	7	25	63
Long Hair	238	27	8	87	263	2	3	92
Carrying Object	182	42	15	121	219	5	21	115

Table 5 presents the values of the parameters of the BLSTM that led the CNN-BLSTM approach to achieve the best results mentioned before. The parameters are sequence length and number of processing units.

Table 5: Parameters that led to obtain the best results for each attribute.

Attribute	Processing Units	Sequence Length
Gender	512	5
Upper Clothes	2048	3
Lower Clothes	32	2
Hat	32	5
Long Hair	128	4
Carrying Object	1024	3

Considering the number of processing units used to obtain the best results for the BLSTM, it seems that there is no clear pattern regarding the issue, since the number of neurons that lead to the best results for each attribute is very heterogeneous: a small number of units such as 32 or 128 led to achieving very satisfactory results, same as higher values such as 1024 and 2048. However, there is a tendency when considering the sequence length: the best performances for half of the attributes were achieved when using large sequence length such as 4 or 5. Hence, the largest the sequence, the better the classifier, which seems coherent since the BLSTM is suitable to treat long-term dependencies. This leads us to consider that longer videos may allow achieving even better results with this method. These conclusions must not be considered as final but instead as initial consideration to foster further studies that should be carried using more datasets and hyper-parameter combinations in order to obtain more conclusive thoughts regarding this issue.

6 Conclusion

This work presented a method to perform soft biometrics classification in videos based on transfer learning and learning temporal dependencies through a recurrent neural network. After detecting an individual in each frame of a video, a pre-trained Convolutional Neural Network was used as feature extractor applied each extracted image. The sequences of features produced by the network were used as input for a Bidirectional Long Short-Term Memory network capable of learning temporal dependencies to perform the classification task. We also pre-processed each frame of the videos by applying a CNN devised for object detection, in order to extract a bounding box containing the individual to be classified. Results showed that learning the temporal variable that underlies a sequence of frames leads to better classification performances than treating a video in a discrete manner, even when using a powerful pre-trained network such as Inception-v3. Since the videos of the dataset used to evaluate our approaches have short length, we can also conclude that recurrent networks such as the BLSTM, which are intended to work with long temporal dependencies, are able to provide satisfactory results even for short sequences of data.

Future works will aim at experimenting with other datasets composed of long-length videos, in order to evaluate if there is an improvement in the performance of our method. Since issues such as partial occlusion or the classification of multiple individuals in a single video were not covered in this work, methods to tackle these particularities can also be developed and evaluated. Multi-class classification, for biometrics such as upper clothes colour (available for the UTFPR-SBD2 dataset), or multi-label classification, in order to classify more than one soft biometric using a single network can be also tested using our method. Individual tracking based on using its soft biometrics attributes could be performed by applying some changes to our method. Finally, it is worth mentioning that the use of soft biometrics classification methods raises some concerns in terms of individual privacy. In this sense, future research could focus not only on technical aspects, such as the development of new algorithms and methods, but also on the study of the social repercussions of their application and the pursuit of fair and responsible use of soft biometrics recognition techniques.

Acknowledgements

M. Romero thanks the Organisation of the American States, the Coimbra Group of Brazilian Universities and the Pan American Health Organization; M. Gutoski would like to thank CAPES for the scholarship; L. T. Hattori thanks CNPQ for the scholarship; H.S.Lopes would like to thank to CNPq for the research grant number 440977/2015-0. All authors would like to thank both CAPES and CNPq for the scholarships, as well as NVIDIA and Fundação Araucária.

References

[1] C. Norris, M. McCahill and D. Wood. “The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space”. *Surveillance & Society*, vol. 2, no. 2/3, 2004.

[2] D. Reid, S. Samangooei, C. Chen, M. Nixon and A. Ross. “Chapter 13 - Soft Biometrics for Surveillance: An Overview”. In *Handbook of Statistics*, edited by C. Rao and V. Govindaraju, volume 31 of *Handbook of Statistics*, pp. 327 – 352. Elsevier, 2013.

[3] A. Dantcheva, P. Elia and A. Ross. “What Else Does Your Biometric Data Reveal? A Survey on Soft Biometrics”. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 441–467, March 2016.

[4] M. Nixon and A. Aguado. *Feature extraction and image processing for computer vision*. Academic press, 2019.

[5] R. O. Duda, P. E. Hart and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[6] Y. LeCun, Y. Bengio and G. Hinton. “Deep learning”. *Nature*, vol. 521, no. 7553, pp. 436–444, 5 2015.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. “You only look once: Unified, real-time object detection”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu and A. C. Berg. “SSD: Single Shot MultiBox Detector”. *CoRR*, vol. abs/1512.02325, 2015.

[9] M. Tan, R. Pang and Q. V. Le. “Efficientdet: Scalable and efficient object detection”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790, 2020.

[10] H. Touvron, A. Vedaldi, M. Douze and H. Jégou. “Fixing the train-test resolution discrepancy”. In *Advances in Neural Information Processing Systems*, pp. 8252–8262, 2019.

[11] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly and N. Houlsby. “Big transfer (BiT): General visual representation learning”. *arXiv preprint arXiv:1912.11370*, 2019.

[12] Q. Xie, M.-T. Luong, E. Hovy and Q. V. Le. “Self-training with noisy student improves imagenet classification”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

[13] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. “Gradient-based learning applied to document recognition”. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] B. L. Hung and H. A. Hung. “Using One-Dimensional Linear Interpolation Method to Check Over-Fitting in Neural Network with Multi-Dimensional Inputs”. In *Frontiers of Manufacturing and Design Science IV*, volume 496 of *Applied Mechanics and Materials*, pp. 2228–2232. Trans Tech Publications, 2014.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna. "Rethinking the inception architecture for computer vision". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[16] A. Krizhevsky, I. Sutskever and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In *Proc. of the 25th International Conference on Neural Information Processing Systems*, volume 1, pp. 1097–1105, USA, 2012. Curran Associates Inc.

[17] C. Szegedy, W. Liu, Y. Jia, P. SermarXivanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich. "Going deeper with convolutions". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Piscataway, NJ, 2015. IEEE press.

[18] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". *arXiv:1409.1556*, 2014.

[19] J. L. Elman. "Finding structure in time". *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[20] S. Hochreiter and J. Schmidhuber. "Long short-term memory". *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*. "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies", 2001.

[22] Y. Nie, C. An, J. Huang, Z. Yan and Y. Han. "A Bidirectional LSTM Model for Question Title and Body Analysis in Question Answering". In *Proc. of the IEEE First International Conference on Data Science in Cyberspace (DSC)*, pp. 307–311, 2016.

[23] L. Shao, F. Zhu and X. Li. "Transfer learning for visual categorization: A survey". *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[24] G. Fung, J. Yu, H. Lu and P. Yu. "Text classification without negative examples revisit". *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 6–20, 2006.

[25] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue and G. Zhang. "Transfer learning using computational intelligence: A survey". *Knowledge-Based Systems*, vol. 80, no. Supplement C, pp. 14 – 23, 2015.

[26] H. A. Perlin and H. S. Lopes. "Extracting human attributes using a convolutional neural network approach". *Pattern Recognition Letters*, vol. 68, pp. 250 – 259, 2015.

[27] G. Levi and T. Hassncer. "Age and gender classification using convolutional neural networks". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pp. 34–42, 2015.

[28] J. Zhu, S. Liao, D. Yi, Z. Lei and S. Z. Li. "Multi-label CNN based pedestrian attribute learning for soft biometrics". In *Proc. of the 2015 IEEE International Conference on Biometrics (ICB)*, pp. 535–540, 2015.

[29] D. Martinho-Corbishley, M. S. Nixon and J. N. Carter. "Retrieving relative soft biometrics for semantic identification". In *Proc. of the 23rd IEEE International Conference on Pattern Recognition*, pp. 3067–3072, 2016.

[30] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye and X. Xue. "Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification". In *Proc. of the 23rd ACM International Conference on Multimedia*, pp. 461–470, 2015.

[31] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici. "Beyond short snippets: Deep networks for video classification". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702. IEEE press, 2015.

[32] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell. "Long-term recurrent convolutional networks for visual recognition and description". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, 2015.

[33] H. Galiyawala, K. Shah, V. Gajjar and M. S. Raval. "Person retrieval in surveillance video using height, color and gender". In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6. IEEE, 2018.

[34] K. He, G. Gkioxari, P. Dollár and R. Girshick. "Mask r-cnn". In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[35] R. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[36] A. Krizhevsky, I. Sutskever and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[37] G. R. Gonçalves, A. C. Nazare, M. A. Diniz, L. E. C. Lima and W. R. Schwartz. "AVSS Challenges 2018 Soft Biometric Retrieval Using Deep Multi-Task Network". In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6. IEEE, 2018.

[38] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger. "Densely connected convolutional networks". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[40] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. PMLR, 2010.

[41] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*, 2014.

[42] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro and et al. "TensorFlow: large-scale machine learning on heterogeneous systems". *arXiv:1603.04467*, pp. 1–19, 2016.