FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 44 (2019) No. 3
ISSN 0867-6356
sciendo DOI: 10.2478/fcds-2019-0017 e-ISSN 2300-3405

Convolutional and Recurrent Neural Networks for Face Image Analysis

Kıvanç Yüksel *, Władysław Skarbek †

Abstract. In the presented research two Deep Neural Network (DNN) models for face image analysis were developed. The first one detects eyes, nose and mouth and it is based on a moderate size Convolutional Neural Network (CNN) while the second one identifies 68 landmarks resulting in a novel Face Alignment Network composed of a CNN and a recurrent neural network. The Face Parts Detector inputs face image and outputs the pixel coordinates of bounding boxes for detected facial parts. The Face Alignment Network extracts deep features in CNN module while in the recurrent module it generates 68 facial landmarks using not only this deep features, but also the geometry of facial parts. Both methods are robust to varying head poses and changing light conditions.

Keywords: deep learning, convolutional neural networks, recurrent neural networks, facial landmark localization, facial parts detection, computer vision, image processing

1. Introduction

Computer vision plays an important role in today's world, from self-driving cars to pedestrian tracking it has many different use cases in many different areas [15]. Face image analysis as a part of computer vision also has its place in plenty divergent real world applications [18]. In this paper two models that are related to face analysis were developed. One of which is a novel algorithm for facial landmarks localization, and the other one is a face parts detector (for eyes, nose and mouth).

The first machine learning system that is presented is a Facial Parts Detector with acronym **CFPD** (Convolutional Facial Parts Detector) . The model is based on

*Promity, Warsaw, Poland, formerly MSc student of Warsaw University of Technology, email: kivanc.yuksel@promity.pl

†Institute of Radioelectronics and Multimedia Technology, Warsaw University of Technology, email: w.skarbek@ire.pw.edu.pl

a moderate size Convolutional Neural Network, and given a face image, it outputs locations of bounding boxes for eyes, nose and mouth. The strength of the presented model, because of the way it was trained, is that it is very robust to difficult face poses. The initial purpose of this model was to reduce the area of search of a face in order to better localize facial landmarks. However it is found that there is no significant improvement when it is used for this goal, thus it is left as a standalone facial parts detector.

The second machine learning system that is presented is a Facial Landmark Localization model with acronym **COREFAN** (Convolutional and Recurrent Face Alignment Network). The reason behind the name is that the model consists of two parts: a traditional convolutional network for feature extraction, and a recurrent network for landmark localization. Although it can be thought as to have two parts, there is a single model which is trained end-to-end. Facial landmark localization is a term used to locate a number of predefined points on a person's face. The purpose of such systems is to find the pixel coordinates of these points when a face image is given as input. Facial expressions recognition [22] , 3D face model extraction [22] , head pose estimation [24] , normalizing face position in the image are some of the use cases of these systems.

The paper is organized in the following manner. In section 2 and 3 detailed descriptions of the proposed methods are provided, and at the end of each of these sections the models were evaluated. Short discussion and comments to existing works considered as the state of art[1], is postponed to the technical sections in order to avoid background redundancy.

The main contributions of this paper are;

1. A novel way of using Convolutional Neural Networks to produce a seeding vector for Recurrent Neural Network type architectures is introduced. This is an uncommon method that is believed to have lots of potential. Furthermore, proposed LSTM architecture is different than usual ones. It shares input weights in the predefined time intervals which is the index interval for landmarks belongs to a facial part.

2. It is shown that image augmentation is a very handy operation that can make trained models more robust to the obstacles that wants to be overcome. To illustrate, three models were trained using original images from the dataset, normalized (to canonical pose) version of these images , and randomly scaled, translated and rotated images. For CFPD, the model trained using augmented images reduced the failure rate by 33% compared to the model trained using original images, and 61% compared to the model trained using normalized images on the IBUG dataset.

[1] In the current fast progress in DNN applications, the time span for being considered as the state of art refers usually to few months.

2. Convolutional Facial Parts Detector – CFPD

In this section a moderate size convolutional neural network model that runs in real time for facial parts (eyes + eyebrows, nose and mouth) detection is presented. The model is an independent model that can run on top of any face detector. Its strong characteristic is that it is very robust to challenging head poses due to the way it was trained. It takes a face image returned by a face detector as its input, and it outputs the bounding boxes for each facial part. Initially the model was trained to reduce the area of search for facial landmarks localization. However, attempts to build an accurate and efficient facial landmarks detector with the help of facial parts detector are failed in all of the cases it was tested. Thus, it was left as a separate detector for real time facial parts detection. Before using a Convolutional Neural Network (CNN) , the detector had been tried to be built using Histograms of Oriented Gradients (HOG) [5]. The main reason for that was to reduce the time complexity of the detection. HOG descriptors extracts the shape information of objects in an image, and it can be trained to detect any object based on its shape. Although it can effectively detect many different kind of shapes such as faces, pedestrians, ships; it is found to be not efficient enough for facial parts detection. The deficiency is not the fault of HOG descriptors, but the insufficiency of shape information to differentiate different parts of the face. This has resulted in many false positives as well as wrongly classified parts (especially for eyes and mouth). Neural networks for computer vision applications provides more natural solutions than the traditional algorithms. Instead of trying to find good features manually, they find convenient features automatically by their own to give better results. Therefore, conventional sliding window approach mostly gives worse results than non-sliding window approach. The abundance of false positives are also the implication of sliding window.

2.1. Data Preprocessing

In order to train the CFPD model, the data released for 300W [14] competition is used. This dataset is collection of images from five datasets (LFPW [2], AFW [24], HELEN [12], 300W [14], and IBUG [14]). Each image in the dataset comes with pixel coordinates of 68 facial landmarks. Having the ground truth landmarks for each image, parts that wants to be detected had been extracted from each image. As the main goal here was to detect facial landmarks later on, eyebrows are also taken together with eyes. Although there are 68 landmarks, for this model only a subset of them are used to locate the top-left, and bottom-right corner of each face part (c.f. Figure 1).

Having ground truth landmarks for each image in the dataset, the images can be easily normalized to a canonical pose using affine transformation to get better accuracy. As a design choice, resolution of the input images is chosen to be 112×112. Each face image in the dataset is scaled to cover the half of the target resolution, translated to be in the center, and rotated to have a canonical pose. For these actions, there are three parameters to be found: the scale coefficient, the rotation angle and

Figure 1. Extraction of ground truth bounding boxes from dataset to be used in training

the translation vector to fill up the transformation matrix:

$$Translation = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix}, \quad Rotation = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Scale = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Hence the transformation matrix (composed firstly from translation, then rotation, and finally of scaling matrices) has the form:

$$T = \begin{bmatrix} cos(\theta)S_x & sin(\theta)S_y & cos(\theta)T_x + sin(\theta)T_y \\ -sin(\theta)S_x & cos(\theta)S_y & -sin(\theta)T_x + cos(\theta)T_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The center of the facial landmarks is set as rotation center. The eye centers is found using a subset of 68 facial landmarks, and the angle between them is taken as the rotation angle (c.f. Figure 2).

The scale parameter is found using the width of the face image. The target shape for network input is 112×112, and the desire is to have the face image's width equal to half of the target shape's width. Lastly, the transition parameter is the difference between rotation center and the target shape's center. The resulting normalized face shape is shown in the figure 2.

After data normalization, the model was trained using a CNN. When the model is tested, it is seen that for the face images that have canonical head pose in the testing set it had much higher accuracy than the ones that do not have. To avoid this, artificial images were added by rotation, scaling, translating, and mirroring of the images from the training set. The parameters were randomly sampled from a normal distribution. The Figure 3 shows examples from the augmented training set.

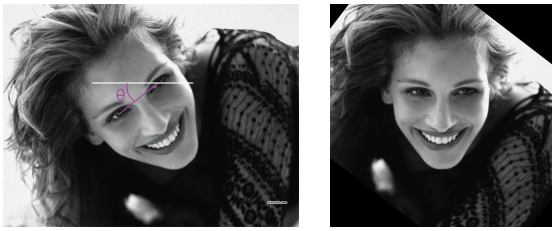

Figure 2. Left: Finding the rotation angle based on eye centers, Right: Normalized face shape

Figure 3. Data augmentation by rotating, scaling and translating

2.2. Algorithm Description

This subsection gives details of the proposed model for facial parts detection. The model takes a face image as input and outputs bounding boxes for each face part. The augmented training set makes the model very robust to difficult head poses and diverse scales. The feed forward structure of the neural network can be seen from the Table 1. Transfer learning was not used in either of the models described in this paper. This means that for both models, network parameters were trained from scratch.

The convolutional layers learn necessary features, and the fully connected layers extracts these features for the defined task. The model outputs 16 dimensional vector where each four dimensions belong to a single part of a face. Each of these four dimensions give upper left and downer-right pixel coordinates of bounding boxes for

Table 1. Feed forward structure of CFPD

Layer	Input Shape	Output Shape	Filter Shape
input	$112 \times 112 \times 1$	$112 \times 112 \times 1$	-
conv-1	$112 \times 112 \times 1$	$112 \times 112 \times 64$	$3 \times 3 \times 1, 1$
conv-2	$112 \times 112 \times 64$	$112 \times 112 \times 64$	$3 \times 3 \times 64, 1$
maxpool-1	$112 \times 112 \times 64$	$56 \times 56 \times 64$	$2 \times 2 \times 1, 2$
conv-3	$56 \times 56 \times 64$	$56 \times 56 \times 128$	$3 \times 3 \times 64, 1$
conv-4	$56 \times 56 \times 128$	$56 \times 56 \times 128$	$3 \times 3 \times 128, 1$
maxpool-2	$56 \times 56 \times 128$	$28 \times 28 \times 128$	$2 \times 2 \times 1, 2$
conv-5	$28 \times 28 \times 128$	$28 \times 28 \times 256$	$3 \times 3 \times 128, 1$
conv6	$28 \times 28 \times 256$	$28 \times 28 \times 256$	$3 \times 3 \times 256, 1$
maxpool3	$28 \times 28 \times 256$	$14 \times 14 \times 256$	$2 \times 2 \times 1, 2$
fc-1	$14 \times 14 \times 256$	$1 \times 1 \times 128$	-
fc-2	$1 \times 1 \times 128$	$1 \times 1 \times 16$	-

different parts. Mean squared error is used as the loss function, and Adam optimizer [8] with initial learning rate 0.01 (which was decayed every 10 epochs by 0.4) was chosen to minimize the mse error:

$$mse = \frac{1}{16} \sum_{n=1}^{16} (y - \hat{y})^2 \tag{3}$$

Although the model was initially trained to reduce the search area for locating facial landmarks, it was not used for this purpose and left as a standalone facial parts detector. The reason for that is (partially) because it did not give any significant improvements compared to the other models. Therefore a better way of aligning facial landmarks, that will be explained later, had to be developed.

2.3. Experiments

Dataset is split into three parts for training, validating, and testing. The validation set was used to monitor the progress of learning, and early stop if the model exhibits overfitting during training. The testing set was also split into three categories to test the accuracy of the model. The first category is called “common set”, and it consists of subset of 300W public testing set, namely testing sets of LFPW and HELEN. The second category is called “challenging set”, as its name refers it consists of challenging images from IBUG dataset. And the last one is 300W private testing set. Some of the outputs of the model is given in the figure 4.

With Nvidia GeForge GTX 1050Ti graphics card, single step with batch size of 64 takes approximately $0.5s$. Training time of the model can be inferred from this according to the number of images that are used. The model was also tested with a camera using Dlib [7] library’s face detector. The fps score of the model can be seen from 3

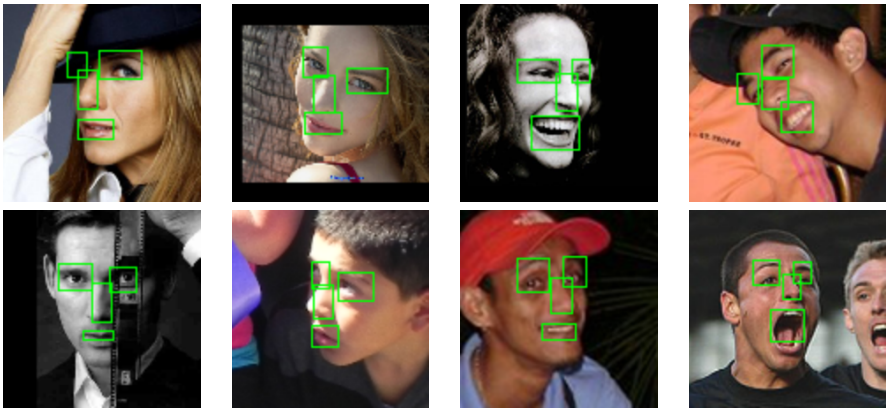

Figure 4. The results of the trained model (CFPD). Up: Randomly sampled images from the testing set, down: Images that the network gave highest errors.

Table 2. Mean squared error for facial parts obtained for the testing set. Three models are trained using original images, normalized images (to canonical pose), and augmented images as described.

	Common Set	Challenging Set	W300 Set
Original images	1.371	4.977	2.986
Normalized images	3.005	8.715	4.832
Augmented images	1.112	3.328	2.604

3. Convolutional and Recurrent Facial Alignment Network – COREFAN

Recurrent Neural Networks (RNN) have been gaining popularity exponentially in recent years. One of the reasons is that not only RNN evaluates its input(s) instantly, but also its evaluation depends on the past input(s). Thus, the output is produced from a collection of information coming from past and present. Its characteristic feature that separates it from the other Neural Network components, is the concept of block memory. This parameterized neural block enables solutions of many different problems such as image captioning [6], language to language machine translation [1], sentiment classification/analysis [16], video classification on frame level [21], and others. The proposed model uses one of the modified RNN architecture that is called Long-Short Term Memory (LSTM).

Table 3. Fps score of the model when it is tested with a camera.

	fps (frames per second)
CFPD	≈ 230
Dlib's face detector	≈ 7
Total	≈ 6.5

3.1. An example of RNN that inspired the basis of COREFAN

In this subsection, an illustrative example of RNN is discussed. This example is important for understanding of the proposed architecture. It is commonly used as a "Hello World" project. The goal of this example is to teach an RNN the sine function. Since we would like the network to learn the sine function, there is no necessity for an external dataset, the data can be computer generated.

The first step is to generate the correct output sequences. Our input to the network will be a subset of the generated sine function in a specific interval l, namely from time step T_t to time step T_{t+l} . Since we want from RNN to generate the next point in time, the correct output sequence for this time step is going to be the sine function sampled from the time step T_{t+1} to the time step T_{t+l+1}.

Explanations after this point are written to enlighten why this example is so important for the proposed face alignment network. Using the trained RNN model, one can generate new sequences of the training data structure by something called "seeding". Instead of inputting a training instance and see what network outputs, any vector can be used as an input and model's response can be investigated. For instance, if we initialize the network with zeros (seed with zeros), and ask from it to generate new sequences of the trained model for a number of time steps, and next join the outputs together, we get something like in the figure 5.

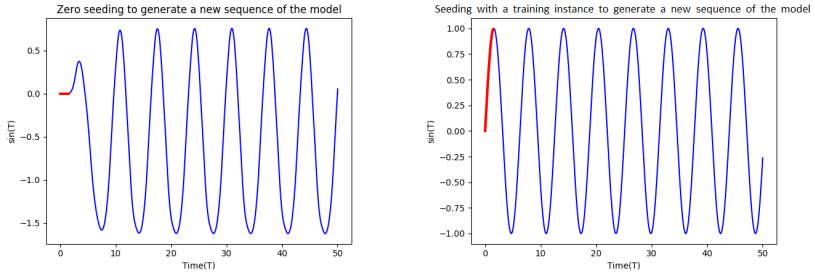

Figure 5. Seeding the trained model with zeros vs with a training instance to generate new sequences

The important point to notice here is that according to the way the network

initialized (seeded) the amplitude, the phase and the frequency of the function[2] is changed. This detail constitute the basis of the designed face alignment network, which will be explained in the following subsection.

3.2. Algorithm Description

The idea of training a face alignment network using RNNs came from the sine example given above (cf. Fig 5). Although RNNs were used to localize facial landmarks before [17, 19], they are mostly used in the refinement process. The way RNNs are used in this project is different than the way it was used in these papers.

In MDM [17] authors also used a convolutional neural network for feature extraction and a recurrent neural network for facial landmarks localization. Their model starts with an initial estimate of the minimum of the energy landscape to produce a series of descent directions that iteratively lead to the optimum. The initial estimate is mean face aligned to the output of a face detector (DAN [9] model also uses initial face estimation. This is a very common method for face alignment. However, the method described in this paper does not use any initial estimate of the landmark coordinates).

In RAR [19] authors also followed the pipeline of cascaded regressions (refining landmarks' locations iteratively). Instead of updating all of the landmarks together, their method instead refines landmarks sequentially using RNN. Authors also introduced two LSTM models namely, attentian LSTM (A-LSTM) and refinement LSTM (R-LSTM). The purpose of A-LSTM is to identify a reliable landmark as the attention center. And the purpose of R-LSTM is to sequentially refining landmarks near or correlated with the attention centers.

Frequency and phase of a sine wave, as function parameters, can change the way a sine wave looks. Also amplitude of a sine wave can be changed by multiplying the function with a constant. These are the parameters that changes the shape of sine function. The idea for the proposed model is: If we stop thinking the facial landmarks as individual points, and think them as building different shapes when the subset of them are grouped together (e.g. landmarks for eyebrows, eyes, mouth, nose and chin), we can train a recurrent neural network to learn these shapes as functions with parameters, and these parameters change the way these shapes look.

The question is: How to find those parameters for each facial part that is chosen? In the sine example it is shown that, when the network learns how sine function works, it produces different shapes of the sine function when it is initialized differently. So initialization is the key here, it acts as function parameters, to produce different shapes of facial parts. Therefore, a convolutional neural network can be trained to extract necessary features from a face image to give the correct initialization values for desired facial shapes for a particular individual.

The facial landmarks are separated into groups of five classes: Chin, eyes, eyebrows, nose and mouth. CNN outputs an initialization (seeding) vector S for each

[2]The frequency modifications make the function more the sine like function, not the original sine function.

class, and a feature vector f_{ih} that contains extracted features. Although there are five different classes/parts, there is not five independent hidden state vectors for each class. Each part continues to use the latest hidden state vector from previous part as the initial hidden state vector of itself. The reason for that is because as well as there is a correlation among the class members, there is one among classes as well. But, how does the RNN know anything about the extracted features of CNN? The feature vector f_{ih} is used to initialize the hidden state vectors. Each face part has additional trainable weights to extract specific features from the general feature vector f_{ih} , thus each part can learn which features are important, combine this information with the latest hidden state of the previous part, and produce a initial hidden state vector for itself. The Figure 6 shows the diagram of the proposed method, and the Table 4 gives the network structure of the CNN part.

The method described in this paper, to the best of our knowledge, is the first system that uses a CNN to produce a seeding vector to an RNN in order to localize facial landmarks.

As like the first model, Adam optimizer [8] was used to train the network. The initial learning rate was set to 0.001 and it was decayed every 20 epochs by 0.5.

Table 4. Feed forward structure of the CNN part of the network (Filter Shape: $(height \times width \times depth, stride)$

Layer	**Input Shape**	**Output Shape**	**Filter Shape**
input	$112 \times 112 \times 1$	$112 \times 112 \times 1$	-
conv-1	$112 \times 112 \times 1$	$112 \times 112 \times 64$	$3 \times 3 \times 1, 1$
conv-2	$112 \times 112 \times 64$	$112 \times 112 \times 64$	$3 \times 3 \times 64, 1$
maxpool-1	$112 \times 112 \times 64$	$56 \times 56 \times 64$	$2 \times 2 \times 1, 2$
conv-3	$56 \times 56 \times 64$	$56 \times 56 \times 128$	$3 \times 3 \times 64, 1$
conv-4	$56 \times 56 \times 128$	$56 \times 56 \times 128$	$3 \times 3 \times 128, 1$
maxpool-2	$56 \times 56 \times 128$	$28 \times 28 \times 128$	$2 \times 2 \times 1, 2$
conv-5	$28 \times 28 \times 128$	$28 \times 28 \times 256$	$3 \times 3 \times 128, 1$
conv-6	$28 \times 28 \times 256$	$28 \times 28 \times 256$	$3 \times 3 \times 256, 1$
maxpool-3	$28 \times 28 \times 256$	$14 \times 14 \times 256$	$2 \times 2 \times 1, 2$
conv-7	$14 \times 14 \times 256$	$14 \times 14 \times 512$	$3 \times 3 \times 256, 1$
conv-8	$14 \times 14 \times 512$	$14 \times 14 \times 512$	$3 \times 3 \times 512, 1$
maxpool-4	$14 \times 14 \times 512$	$7 \times 7 \times 512$	$2 \times 2 \times 1, 2$
fc-1	$7 \times 7 \times 512$	$1 \times 1 \times 512$	-
fc-2	$1 \times 1 \times 512$	$1 \times 1 \times 50$	-

The way the feature vector f_{ih} is included in hidden state calculations is as follows,

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1},\ S_{part}] + b_f), \qquad i_t = \sigma(W_i \cdot [h_{t-1},\ S_{part}] + b_i) \\
\widetilde{C}_t &= tanh(W_C \cdot [h_{t-1},\ S_{part}] + b_C), \qquad C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1},\ S_{part}] + b_o), \qquad h_t = o_t * tanh(C_t + W_{ih_n} \cdot f_{ih} + b_{ih_n})
\end{aligned}
\tag{4}
$$

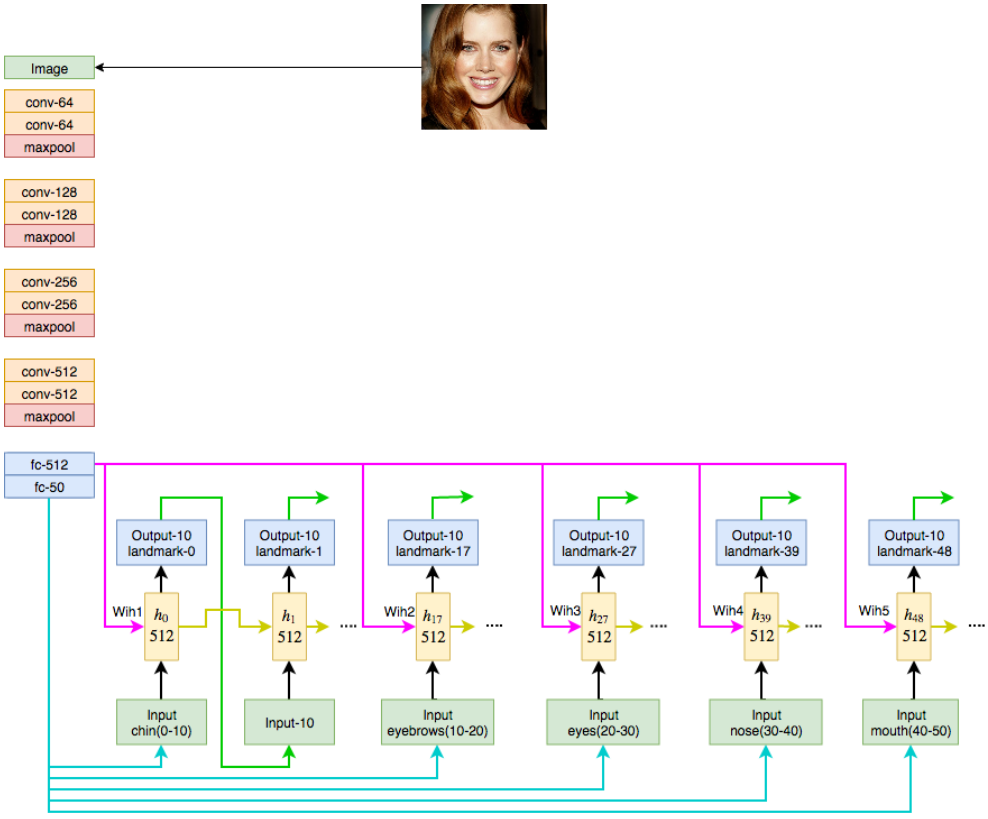

Figure 6. A diagram illustrating the idea of proposed method

Where S_{part} is the seeding vector for the face parts (chin, eyebrows, eyes, nose and mouth), f_{ih} is the feature vector extracted by the CNN, and the W_{ih_n} and b_{ih_n} trainable weights and biases for the parts[3].

Lastly, to be consistent with other face alignment methods [23] [13] [20] and to be able to compare the results, the loss function that was used for the network is landmark location error normalized by the distance between pupils.

3.3. Data Preprocessing

For training the model the same dataset that was used for CFPD (section 2) is used, and the same data preprocessing steps are followed. The data provided by 300W

[3]The proposed RNN architecture simulates the time via landmark index. It is presented in an unfolded form ready to gradient flow at the backward error propagation. However, it differs from the standard LSTM architectures by sharing input weights in the predefined time intervals. Here the time interval is the index interval for landmarks included in the same facial part.

competition comes with the locations of predefined ground truth landmarks. However these landmarks have to be processed in order to fit our needs. As it was said the landmarks are grouped into five sets to build different face parts. Every subset of landmarks for each face part then needs to be processed to build a training sequence.

The seeding parameters for each facial part is coming from the output of CNN. For sequence length l, CNN outputs $5 \times l$ dimensional vector where each length of size l corresponds to seeding parameters for a single facial shape. Since at each time step we would like to get a single landmark coordinate, the input and output sequence of the model should share some common values. To illustrate the concept, if we assume that there are only 10 facial landmarks in each face part, and $l = 4$, figure 7 show how the output sequence for training was prepared.

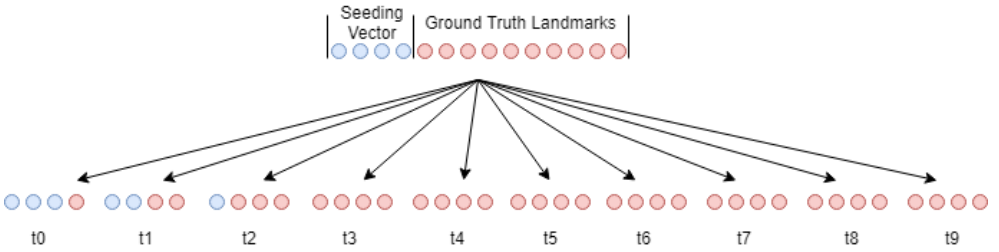

Figure 7. Data preprocessing example for a smaller set of ground truth landmarks

3.4. Experiments

In this section extensive evaluation of described model is performed on the same dataset that was used to evaluate CFPD in subsection 2.3. To test the model two error measures that was used by other face alignment networks were used:

- the mean distance between the network output and ground truth landmarks normalized by the inter-ocular distance,
- the mean distance between the network output and ground truth landmarks normalized by the inter-pupil distance.

Apart from the error measures above, as some of the other models did [9] , area under the cumulative error distribution curve AUC_α and failure rate of the model is also tested. The AUC_α is calculated as the area under the cumulative distribution curve calculated up to a threshold α, then divided by that threshold, and each image with an inter-ocular normalized error of 0.08 or greater considered to be a failure. The results are shown in the Table 5 and 7. The separation of the testing set is exactly the same as it was explained in the subsection 2.3. Lastly, figure 8 shows results on the randomly selected images from the challenging subset, as well as 4 worst results based on inter-ocular error.

Table 5. Comparison of normalized error of different face alignment methods on the 300-W public testing set

Method	Common Set	Challenging Set	Full Set
inter-pupil normalization			
SDM [20]	5.60	15.40	7.52
ESR [3]	5.28	17.00	7.52
LBF [13]	4.95	11.98	6.32
CFSS [23]	4.73	9.98	5.76
Kowalski et al. [10]	4.62	9.48	5.57
DAN [9]	4.42	7.57	5.03
DAN-Menpo	4.29	7.05	4.83
RAR [19]	4.12	8.35	4.94
cGPRT [11]	-	-	5.71
COREFAN (Original images)	5.42	12.15	6.73
COREFAN (Normalized images)	7.35	16.74	9.19
COREFAN (Augmented images)	4.87	9.24	5.72
inter-ocular normalization			
MDM [17]	-	-	4.05
Kowalski et al. [10]	3.34	6.56	3.97
DAN [9]	3.19	5.24	3.59
DAN-Menpo	3.09	4.88	3.44
COREFAN (Original images)	3.91	8.40	4.78
COREFAN (Normalized images)	5.30	11.60	6.53
COREFAN (Augmented images)	3.51	6.40	4.07

With Nvidia GeForge GTX 1050Ti graphics card, single step with batch size of 64 takes approximately $0.76s$. Training time of the model can be inferred from this according to the number of images that are used.

4. Conclusions

In this paper, two models for face image analysis are presented. The first model is a moderate size Face Parts Detector (eye+eyebrows, nose, mouth) that is based on a convolutional neural network – CFPD, and the second one is a novel Face Alignment Network that is a combination of a convolutional and a recurrent neural network – COREFAN.

The first detector takes a face image as input, and it outputs bounding boxes for

Table 6. Comparison of AUC_α and failure rate of different face alignment methods on the 300-W public testing set

Method	$AUC_{0.08}$	Failure(%)
inter-ocular normalization		
SDM [20]	42.94	10.89
ESR [3]	43.12	10.45
CFSS [23]	49.87	5.08
MDM [17]	52.12	4.21
DAN [9]	55.33	1.16
DAN-Menpo [9]	57.07	0.58
COREFAN (Original images)	51.27	0.54
COREFAN (Normalized images)	38.09	13.35
COREFAN (Augmented images)	56.10	0.18

Table 7. Comparison of mean error, AUC_α and failure rate of different face alignment methods on the 300-W private testing set

Method	Mean Error	$AUC_{0.08}$	Failure(%)
inter-ocular normalization			
MDM [17]	5.05	45.32	6.80
DAN [9]	4.30	47.00	2.67
DAN-Menpo [9]	3.97	50.84	1.83
ESR [3]	-	32.35	17.00
CFSS [23]	-	39.81	12.30
COREFAN (Original images)	5.84	32.61	13.5
COREFAN (Normalized images)	7.58	23.52	30.16
COREFAN (Augmented images)	5.12	39.32	6.16

each face part. It can be used on top of any face detector, and its strength is that it is, because of the way it was trained, very robust to difficult face poses. The model is tested on the data provided for the 300W competition, and the results are very satisfying. Even for the challenging subset of the testing set, it gives 3.328 mean square error.

Recurrent Neural Networks' characteristic feature is to have a memory, and use this memory to processing their inputs according to the past information. This allows very interesting use cases of this type of neural networks. In this paper, a variation of RNN that is called LSTM is used for face image analysis, and a novel idea of using convolutional neural networks to produce a seeding vector for RNNs is presented. Moreover, the proposed architecture it differs from the standard LSTM by sharing input weights in the predefined time intervals. Here the time interval is the index interval for landmarks included in the same facial part. The unfolded architecture makes the gradient flow at the backward error propagation fully specified, contrary to the traditional architecture unfolding not during the design but during the training stage.

Figure 8. The results of the trained model (COREFAN). Up: Randomly sampled images from the challenging subset, down: Images with highest error based on inter-ocular normalization.

It is shown that if a convolutional neural network is used to extract features from a face image in order to produce a seeding vector, an LSTM like network can be used to recognize shapes of different face parts, and it can adjust these shapes accordingly for different people and head poses.

We admit that the invariance of the proposed RNN method with respect to varying pose and light conditions was confirmed only by subjective perceptual tests.

Acknowledgment

This work was partially co-financed by the National Centre for Research and Development in Poland funds as part of the project POIR.01.01.01-00-0800/17 developed by the Promity, Warsaw, Poland.

References

[1] Bahdanau, D., Cho, K. and Bengio, Y. 2014, 'Neural machine translation by jointly learning to align and translate', arXiv preprint arXiv:1409.0473 .

[2] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J. and Kumar, N. 2013, 'Localizing parts of faces using a consensus of exemplars', IEEE transactions on pattern analysis and machine intelligence 35(12), 2930–2940.

[3] Cao, X., Wei, Y., Wen, F. and Sun, J. 2014, 'Face alignment by explicit shape regression', International Journal of Computer Vision 107(2), 177–190.

[4] Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

[5] Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[6] Karpathy, A. and Fei-Fei, L. 2015, Deep visual-semantic alignments for generating image de- scriptions, in 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 3128–3137.

[7] King, D.E., 2009. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10(Jul), pp.1755-1758.

[8] Kingma, D. P. and Ba, J. 2014, 'Adam: A method for stochastic optimization', arXiv preprint arXiv:1412.6980 .

[9] Kowalski, M., Naruniec, J. and Trzcinski, T. 2017, 'Deep alignment network: A convolutional neural network for robust face alignment', CoRR abs/1706.01789. URL: http://arxiv.org/abs 1706.01789

[10] Kowalski, M. and Naruniec, J. 2016, 'Face alignment using k-cluster regression forests with weighted splitting', IEEE Signal Processing Letters 23(11), 1567–1571.

[11] Lee, D., Park, H. and Yoo, C. D. 2015, Face alignment using cascade gaussian process regression trees, in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 4204–4212.

[12] Le, V., Brandt, J., Lin, Z., Bourdev, L. and Huang, T. S. 2012, Interactive facial feature localization, in 'European Conference on Computer Vision', Springer, pp. 679–692.

[13] Ren, S., Cao, X., Wei, Y. and Sun, J. 2014, Face alignment at 3000 fps via regressing local binary features, in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 1685–1692.

[14] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S. and Pantic, M. 2013, 300 faces in-the-wild challenge: The first facial landmark localization challenge, in 'Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on', IEEE, pp. 397–403.

[15] Sebe, N. and Lew, M.S., 2013. Robust computer vision: Theory and applications (Vol. 26). Springer Science & Business Media.

[16] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. and Potts, C. 2013, Recursive deep models for semantic compositionality over a sentiment treebank, in 'Proceedings of the 2013 conference on empirical methods in natural language processing', pp. 1631–1642.

[17] Trigeorgis, G., Snape, P., Nicolaou, M. A., Antonakos, E. and Zafeiriou, S. 2016, Mnemonic descent method: A recurrent process applied for end-to-end face alignment, in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 4177–4187.

[18] Wen, Z. and Huang, T.S., 2006. 3D Face Processing: Modeling, Analysis and Synthesis (Vol. 8). Springer Science & Business Media.

[19] Xiao, S., Feng, J., Xing, J., Lai, H., Yan, S. and Kassim, A. 2016, Robust facial landmark detection via recurrent attentive-refinement networks, in 'European conference on computer vision', Springer, pp. 57–72.

[20] Xiong, X. and De la Torre, F. 2013, Supervised descent method and its applications to face alignment, in 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 532–539.

[21] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. and Toderici, G. 2015, Beyond short snippets: Deep networks for video classification, in 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4694–4702.

[22] Yuksel, K., Chang, X. and Skarbek, W., 2017, August. Smile detectors correlation. In Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2017 (Vol. 10445, p. 104451L). International Society for Optics and Photonics.

[23] Zhu, S., Li, C., Change Loy, C. and Tang, X. 2015, Face alignment by coarse-to-fine shape searching, in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 4998–5006.

[24] Zhu, X. and Ramanan, D., 2012, June. Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2879-2886). IEEE.

This paper is a revised and extended version of work originally presented at the 16th International Symposium New Trends in Audio and Video - NTAV2018, 11-13 October 2018, Poznań, Poland

Received 29.01.2019, Accepted 23.04.2019