Facial Key points Detection using MobileNetV2 Architecture

Uday Kulkarni
KLE Technological University
Hubballi,India
uday_kulkarni@kletech.ac.in

Sunil V. Gurlahosur
KLE Technological University
Hubballi,India
svgurlahosur@kletech.ac.in

Pooja Babar
KLE Technological University
Hubballi,India
babarpooja2002@gmail.com

Shreedhar I Muttagi
KLE Technological University
Hubballi,India
muttagi.shreedhar@gmail.com

N Soumya
KLE Technological University
Hubballi,India
soumyan29102002@gmail.com

Priya A Jadekar
KLE Technological University
Hubballi, India
priyajadekar@gmail.com

Dr.Meena S M
KLE Technological University
Hubballi,India
msm@kletech.ac.in

***Abstract*—With the popularity of social media like Instagram and Snapchat, facial filters or beautifying filters are used more often. These applications cannot store raw images of faces when used every time. Thus, they need unique characteristics of a face to which the filters can be applied. These unique characteristic points of every face are called facial key points. With the increasing use of such applications, facial key point detection has become a popular topic. The objective of key point detection is to extract the coordinates of the unique points in the face which are necessary and sufficient to detect the image. Every person has a different face and the key point coordinate for each is very different from the other. Thus, detecting the key points becomes a difficult task. The detection becomes more challenging depending on the angle with which the image is taken and the light exposure of the image. In this paper, we propose the use of deep learning architecture, MobileNet version2 (MobileNetV2) [26] to solve this complex problem as it proves to be better than the traditional architectures available. Our aim is to detect 15 key points of the given facial image using CNN with MobileNetv2 architecture to obtain lower loss and better accuracy. The baseline model used is a single hidden layer neural network and convolutional the advanced model is a Convolutional Neural Network with MobileNetV2 architecture. The experimental results have shown 84% accuracy as compared to the present state-of-art algorithms.**

***Index Terms*—Key points, Baseline Model, Neural Networks, Deep Learning, Convolutional, Coordinates, MobileNetV2**

I. Introduction

Artificial Intelligence [13] is an advancing domain that enables a machine to analyze a given problem like a human brain. Considering how the human brain considers millions of features to solve a problem and reach a rational decision, enabling the same to a machine is a difficult job, in which science has not yet been able to achieve complete success efficiently. One factor which machines have in aWe aimnd that humans can't compete with is computational power. Thus, making use of this high computational power and programming the machine to choose its own given any combination of situations is a way to allow machines to behave similarly to humans. The question that arises now, is why we need artificial intelligence if they are not as rational as humans. The answer to that is to monitor conditions that cannot be done by humans efficiently and to save time. Machine learning is a type of artificial intelligence that allows the machine to learn without specifically being programmed. This involves providing various combinations of possibilities for the same problem and allowing the machine to predict the possible outcome depending on the training data set. Deep learning [14] is a type of machine learning that enables the machine to work similarly to a human brain and select the features that are necessary to train for better efficiency.

The most basic work that any machine does is the recognition of various objects present in the real world, known as object detection [15]. We can find a face in every corner in a widely populated world of 8 billion. There are a lot of applications that make use of facial features to work or provide security. Thus, the recognition of facial key points, which are the deciding factors in face recognition or any other application that uses filters on the face is a key job. Mane and Shah [11] provide insight into all the algorithms in use for facial key points detection. But, every image that is given as input has different shadows, and light angles because it is different in the angle that is taken to click the picture. Thus, processing the image is an important step in facial key point detection to make the training easier. Baffour and others [12] give a survey of algorithms that provide Deep Learning, Deep Neural Networks (DNN)[2] solutions for keypoint detection and also give the problems detected in each of them. The key points include the corners of the eye, the mouth, the nose, and the eyebrows. These are the features that determine the face of a person. Also, just the facial key point storage is sufficient for recognition and other operations, this decreases the data storage space greatly and helps in increasing the robustness of the model.

The traditional methods to extract features such as Local Binary Pattern (LBP) [5], Gabor filters [6], and Histogram of Oriented Gradients (HOG) [7] are commonly used and provide good results. But the problem with these algorithms is that their performance when in competitive use is low. Thus, they

become redundant for applications where key point detection cannot take much time.

The latest approaches to this problem are directed toward neural networks. Jung and others [8] have proposed the use of fine-tuning in neural networks that extract temporal appearance features from image sequences and facial landmark points. They have obtained better results, but the accuracy drops as the number of key points under consideration increases. Mollahosseini [8] has proposed a deep neural network architecture to solve the FER problem and extract deeper features. Shin [9] has given a baseline convolutional neural network which gives the best performance in expression extraction. The use of DCNN(Deep Convolutional Neural Network) has for sure proved to be efficient and accurate, but when the background contains noise, a lot of unnecessary features are extracted which makes the model heavy due to a lot of parameters being calculated.

In this paper, we propose a model that determines the facial key points of the input image with the help of the MobilenetV2 [10] algorithm. We use raw images to detect the key point coordinates with respect to the left corner of the image, after pre-processing it. The architecture of the model is shown in Fig 1.

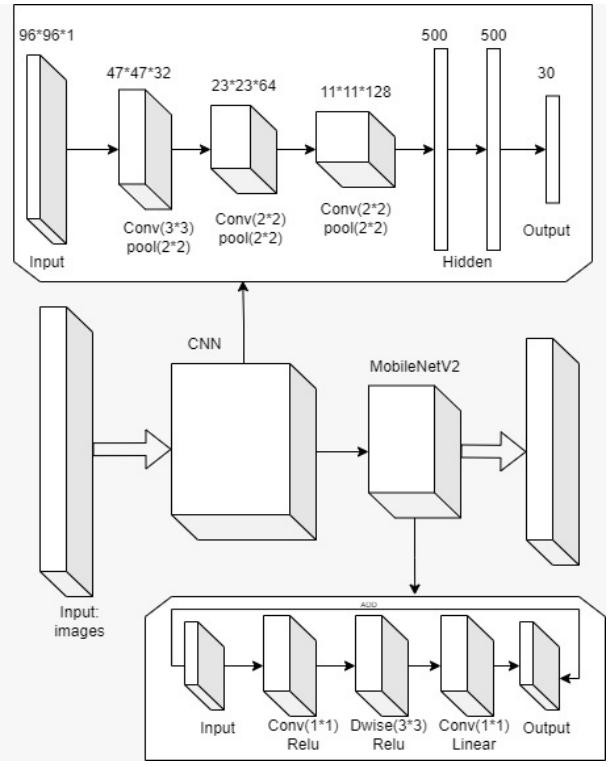

Fig. 1. Architecture of the proposed system

This paper is organized as follows. Section 2 gives the background i.e. the state-of-art and currently implemented systems for the problem statement. In section 3, you will find the explanation for the proposed system followed by the experimental results in section 4. The conclusion for the paper is given in section 6. Lastly, section 7 consists of references.

II. BACKGROUND

The various state-of-art mechanisms used to detect the key points use the raw images or the pixel values as inputs and various networks to provide the results. While Bledsoe was the first to attempt semi-automated face recognition and Bell Labs developed a vector of 21 features, they were quite difficult to automate. One of the earliest successful automated facial recognition systems is Mathew's [1] eigenfaces system. He used the information theory approach of coding and decoding face images. He calculated eigenvalues and images that were corresponding to the highest eigenvalues were called eigenfaces. The features used were weights calculated based on input image and eigenfaces. But the model is not robust and the huge in size.

Naimish [2] proposed architecture in 2017, consisting of 4 convolution2d layers, 4 max-pooling layers, and 3 dense layers, with sandwiched dropout and activation layers that use Exponential Linear Units. His architecture is inspired by LeNet, which uses the Kaggle dataset of facial key points.

Shenhao [3] has tried to use basic machine-learning algorithms and work with python. He has compared the root mean square errors (RMSEs) of every basic algorithm in keypoint detection, which shows CNN has the least error.

Shutong Zang and Chenyue Meng [4] have used a hidden layer and a convolutional layer as their baselines alongside pre-trained inceptions models which are used to enhance the performance of the model. The main aim of this paper was to reduce the time complexity to achieve real-time key points detection. The RMSE loss is used for analyzing the performance. The results show that the Simple Inception model gives a loss of 3.88 which is the worst case, and the Inception CNN model with a loss of 1.13 the best case but does have the drawback of overfitting the dataset.

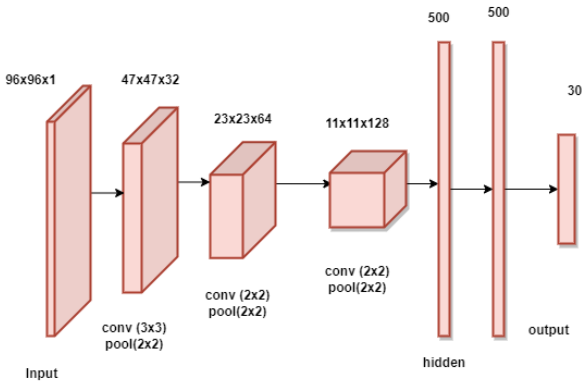

Fig. 2. CNN architecture [26]

This paper proposes a Mobilenetv2 architecture as the

advanced baseline model to obtain better accuracy. The CNN architecture is shown in Fig. 2. The size of the layer's activation is depicted by the number on top of each layer. The number below the layers depicts the type.

III. PROPOSED SYSTEM

A. Methodology

To make a comparative study, we use the single hidden layer neural network for the detection first. As input for the network, we have reshaped the 96 x 96 image to a 9216 x 1 vector. We have added 100 neurons in the hidden layer and the output layer is set to give 15 key points. Thus, the total number is 30. The Euclidean distance between the ground truth and output key points vectors is defined as the loss function. We have used Nesterov momentum to update the gradient descent. We have taken 1 batch size for 400 epochs. Shuffle has been used to make sure every item has the same chance to occur in any position. The accuracy obtained is 79.9%. For better accuracy, we have moved to a convolutional neural network.

TABLE I
HYPER-PARAMETERS OF CNN

	Conv-layer 1		Conv-layer 2		Conv-layer 3	
	conv	pool	conv	pool	conv	Pool
Filter	3 ×3	2 ×2	2 ×2	2 ×2	2 ×2	2 ×2
Pad	0	0	0	0	0	0
Stride	1	2	1	2	1	2

B. MobileNetV2 for facial keypoint detection

The method we propose is called transfer learning where the model learns from the pre-existing model. The architecture for facial key points detection used in the CNN model is MobilenetV2 which is used to extract features from the input facial images. The pre-trained model is imported from Keras. application. The image dataset is augmented and then passed for training. The model uses Adam[28] as an optimizer. After running for 400 epochs, the accuracy obtained is 84%.

An excellent feature extractor for object recognition and segmentation is MobileNetV2. Over the full latency range, the models are quicker for the same accuracy. For many visual identification tasks, this architecture offers a highly effective mobile-oriented approach that can be the foundation.

This architecture is what we employ for efficient keypoint extraction. We load a network that has already been trained, and we use the ImageNet database's millions of photos to train it [29]. Transfer training refers to the process in which a model learns from an existing model.

IV. EXPERIMENTAL RESULTS

A. Experimental Setup

We have conducted experiments on 2 models in total. The first is the single hidden layer network model, which is our baseline model. The second is the CNN model which is our advanced baseline.

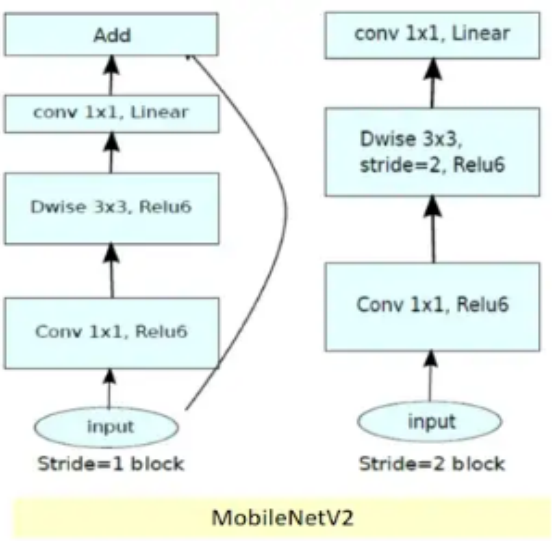

Fig. 3. Architecture of MobilenetV2 [30]

Both experiments were conducted on AMD Ryzen5 5500U with Radeon Graphics + 2.10GHz CPU, 8GB memory laptop. The frameworks used are TensorFlow and imguag for image augmentation.

B. Dataset Description

We use the Kaggle dataset of facial key points detection competition. The total number of 96 x 96 pixels images in the data is 7096. The data also contains four CSV files namely the training, testing, Idol, equitable, and sample submission. Thus, the data is divided into a training set, testing set, and validation set, X_{tr} , X_{tst} , X_{valid} . The corresponding sets to these are Y_{tr} , Y_{tst} , Y_{valid} The X_{tr} represents the given images, in the form of pixels and Y_{tr} . gives the key point for the corresponding image in vector form.

Let our model be M. We train M on (X_{tr} , Y_{tr} .) and tune the parameters. After training, we evaluate M on (X_{tst} , Y_{tst} .).

The dataset contains null values as we can see in the below count graph, null values in the columns including left eyebrow inner end x, left eyebrow inner end y, left eyebrow outer end y, left eyebrow outer end x and others total of 10+ columns contain the null values. Pre-processing of the data has been made by dropping the null values.

Some of the images in the dataset are shown in Fig 5. The test dataset is of size 1783 rows and 2 columns, ImageId, and Image. Out of 7960 images, 2140 have ground truth positions for all 15 keypoints. 80% of the images (1360) are used for training and 20%(440) are used for testing.

C. Evaluation Metrics

The problem is framed the same as the regression model where the the target variable is a continuous numeric variable,

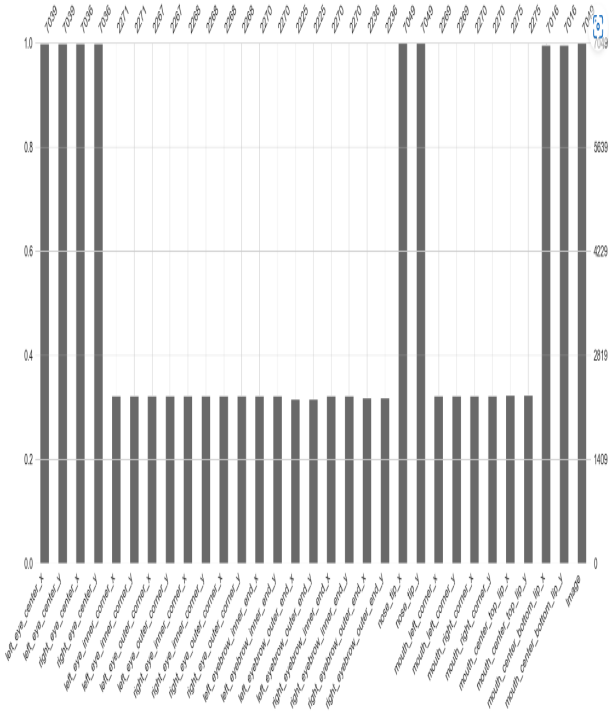

Fig. 4. Countplot for 15 facial key-points

TABLE II
ALL 15 FACIAL KEYPOINTS

Facial Key-points	
1	Left eye center
2	Left eyebrow outer end
3	Right eye outer corner
4	Right eye center
5	Right eyebrow inner end
6	Left eyebrow inner end
7	Left eye inner corner
8	Right eyebrow outer end
9	Mouth right corner
10	Left eye outer corner
11	Right eye inner corner
12	Nose tip
13	Mouth center top lip
14	Mouth left corner
15	Mouth center bottom lip

we here used the mean squared error and the mean accuracy error(mae) to calculate the evulation metrics.

Consider the given ground truth vectors as a $= \{a_0, \ldots\ldots, a_i, \ldots\ldots a_n\}$ and let our estimated vectors be $\hat{a} = \{\hat{a}_0, \ldots\ldots, \hat{a}_i \ldots\ldots, \hat{a}_n\}$.

Mean squared error In machine learning the mean square

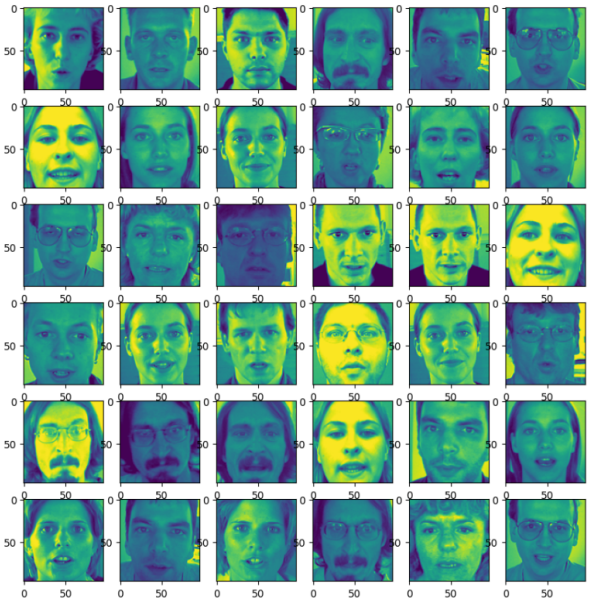

Fig. 5. Dataset images

error is basically used to check the performance of the regression model

$$\text{loss}_{MSE} = 1/n \sum_{i=1}^{n} (a_i - \hat{a}_i)^2 \tag{1}$$

Mean absolute error It is a standard evaluation metric. It calculates the difference between the actual and calculated values predicted by the machine learning model.

$$\text{loss}_{MAE} = 1/n \sum_{i=1}^{n} (|a_i - \hat{a}_i|)^2 \tag{2}$$

The coordinates of the key points are in the range [0.95] x [0,95] and we rescale the coordinates to [-1,1] x [-1,1].
The loss of the single hidden layer network is given in Table 3. The loss of the convolutional neural network with MobilenetV2 is given in Table 4.

TABLE III
LOSS OF SINGLE HIDDEN LAYER

Iter $/10^{-3}$	Single hidden layer		
	Train	Test	Valid
100	6.25	4.32	6.27
200	3.64	2.39	5.09
400	2.50	3.89	3.98
1000	1.09	2.21	2.53

As we can see in table 3 when 100 epochs are given a loss of 6.25 is observed in the training dataset, a 4.32 loss in the test dataset, and a 6.37 loss in the validation dataset. At 1000 epochs a huge reduction in loss rate is observed. To get more accuracy we continued our work with a convolutional

neural network (CNN) with mobilenetv2 architecture (fig 6) by comparing it with a single hidden layer and CNN we get more accuracy in the CNN also a reduction in loss is observed.

TABLE IV
LOSS OF CNN WITH MOBILENETV2

Iter $/10^{-3}$	CNN with MobileNetV2		
	Train	test	Valid
100	5.03	4.21	4.32
200	1.50	3.54	3.23
400	1.09	3.21	2.32
1000	.95	1.13	1.15

For the implementation of the proposed methodology, we have used MobileNetV2 with CNN by increasing the epochs to 1000 and with a learning rate of 0.8 we get more accuracy.

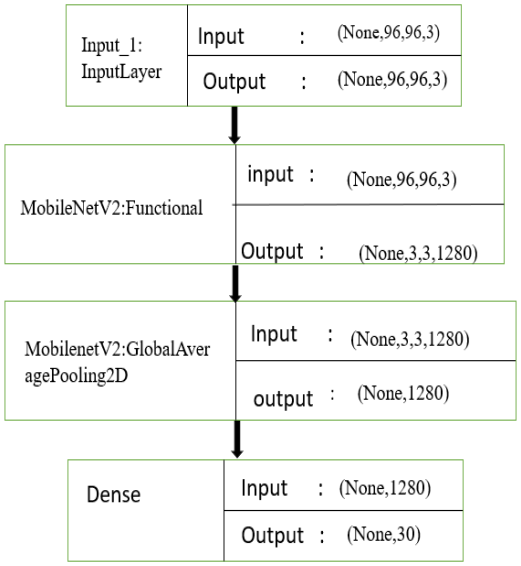

Fig. 6. MobileNetV2 architecture of purposed model

D. Results

We have evaluated our model after 400 epochs and the accuracy it has provided is 78%. The key points detected are shown in Fig 9. The comparative study of the losses of the single hidden layer and CNN with MobileNetv2 network is shown in Table5.

To address the regression issue, transfer learning architectures like MobileNetV2 have been modified. Here, a GAP + Regression (Dense Layer) has been used to replace the topmost Softmax classification's original weights in order to forecast the facial key-points. By completely fine-tuning all the layers, the models are tested using Imagenet's original baseline weights.

TABLE V
LOSS OF BOTH MODELS

Iter $/10^{-3}$	Single hidden layer			CNN with MobileNetV2		
	train	test	Valid	Train	test	Valid
100	6.25	4.32	6.27	5.03	4.21	4.32
200	3.64	2.39	5.09	1.50	3.54	3.23
400	2.50	3.89	3.98	1.09	3.21	2.32
1000	1.09	2.21	2.53	.95	1.13	1.15

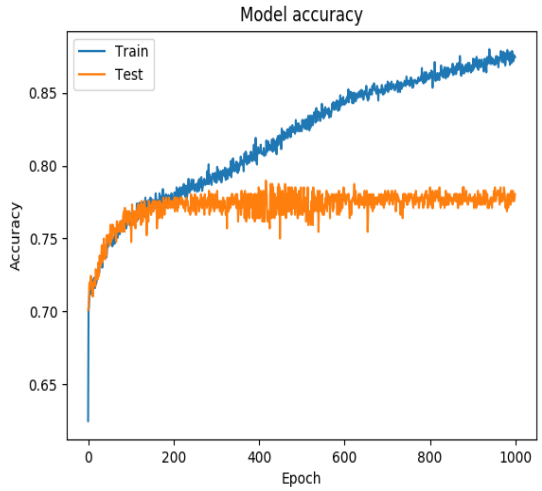

Fig. 7. CNN Model accuracy vs number of epochs

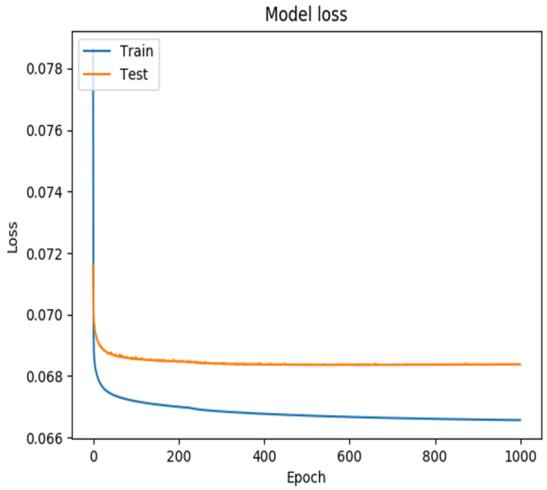

Fig. 8. CNN Model loss vs number of epochs

The temporal complexity is a crucial test metric for the model. Table 6 presents the outcomes. Testing time is the cost time for predicting the important parts of one test image, while training time is the cost for one epoch (1700 images). Consequently, there were roughly 106 multiplication operations in total (ignoring the impact of additions).

TABLE VI
TIME CONSUMPTION IN TRAINING AND TESTING

	Single hidden layer	CNN with MobileNetv2
train /s	0.29	39
test /ms	0.102	8.98

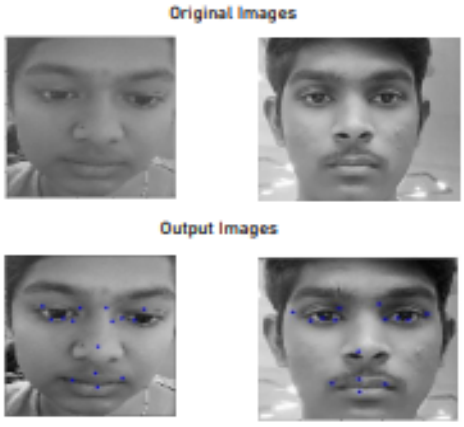

Fig. 9. Input and Output Images

V. CONCLUSION

In this paper, we have proposed a convolutional model with MobilenetV2 for the detection of 15 key points as given in the Kaggle dataset for a given input image. The experimental results show that the model is better than most of the state-of-art models with 84% accuracy and an RMSE loss of around 1.2. However, the time complexity of the CNN model is 39 seconds, which makes the model slower in real-world scenarios.

As for our future work, we can explore more on adding the Inception model to the CNN model to decrease the overfitting problem which may occur in complex real-world scenarios. Also, it will help in image recognition proficiency when passed through the training data set several times.

REFERENCES

[1] M.Turk & A.Pentland: Eigenfaces for recognition, J.Cog.Neuroscience, Volume 3, (1991)

[2] Naimish Agarwal, Artus Kohn-Grimberghe and Ranjan Vyas, Facial Key Points Detection using Deep Convolutional Neural Network – NaimishNet, (2017)

[3] Shenghao Shi, Facial Keypoints Detection (2017) (mathpix)

[4] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on local binary patterns: A comprehensive study," Image and vision Computing, vol. 27, no. 6, pp. 803-816, 2009.

[5] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," IEEE Transactions on Image processing, vol. 11, no. 4, pp. 467-476, 2002.

[6] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, "Disfa: A spontaneous ESAET 2020Journal of Physics: Conference Series 1549 (2020) 022136 IOP Publishingdoi:10.1088/1742-6596/1549/2/0221367 facial action intensity database," IEEE Transactions on Affective Computing, vol. 4, no. 2,pp. 151-160, 2013.

[7] H. Jung, S. Lee, J. Yim, S. Park and J. Kim, "Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2983-2991, doi: 10.1109/ICCV.2015.341.

[8] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in 2016 IEEE Winter conference on applications of computer vision (WACV), 2016: IEEE, pp. 1-10.

[9] M. Shin, M. Kim, and D.-S. Kwon, "Baseline CNN structure analysis for facial expression recognition," in 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2016: IEEE, pp. 724-729.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C.Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.

[11] A. Srivastava, S. Mane, A. Shah, N. Shrivastava and B. Thakare, "A survey of face detection algorithms," 2017 International Conference on Inventive Systems and Control (ICISC), 2017, pp. 1-4, doi: 10.1109/ICISC.2017.8068607.

[12] Awuah Baffour, Prince & Nunoo-Mensah, Henry & Keelson, Eliel & Kommey, Benjamin. (2022). A Survey on Deep Learning Algorithms in Facial Emotion Detection and Recognition. 7. 24-32. 10.25139/inform.v7i1.4282.

[13] Alsedrah, Mariam. (2017). Artificial Intelligence. 10.13140/RG.2.2.18789.65769.

[14] LeCun, Yann& Bengio, Y.& Hinton, Geoffrey. (2015). Deep Learning. Nature. 521. 436-44. 10.1038/nature14539.

[15] Amit, Yali & Felzenszwalb, Pedro & Girshick, Ross. (2020). Object Detection. 10.1007/978-3-03003243-2660-1.

[16] Lida, Hu & Ge, Qi. (2020). Automatic facial expression recognition based on MobileNetV2 in Real-time. Journal of Physics: Conference Series. 1549. 022136. 10.1088/1742-6596/1549/2/022136.

[17] Kulkarni, Uday, S. M. Meena, Sunil V. Gurlahosur, and Gopal Bhogar. "Quantization Friendly MobileNet (QF-MobileNet) architecture for vision based applications on embedded platforms." Neural Networks (2020).

[18] U. Kulkarni, S. M. Meena, S. V. Gurlahosur and U. Mudengudi, "Classification of Cultural Heritage Sites Using Transfer Learning," 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), 2019, pp. 391-397, doi: 10.1109/BigMM.2019.00020.

[19] U. Kulkarni, S. M. Meena, P. Joshua, K. Rodrigues and S. V. Gurlahosur, "Integrated Crowdsourcing Framework Using Deep Learning for Digitalization of Indian Heritage Infrastructure," 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), New Delhi, India, 2020, pp. 200-208, doi: 10.1109/BigMM50055.2020.00036.

[20] U. Kulkarni, S. M. Meena, P. Joshua, K. Rodrigues and S. V. Gurlahosur, "Integrated Crowdsourcing Framework Using Deep Learning for Digitalization of Indian Heritage Infrastructure," 2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM), New Delhi, India, 2020, pp. 200-208, doi: 10.1109/BigMM50055.2020.00036.

[21] Uddin, Md & Barman, Paresh & Ahmed, Khandaker & Rahim, S & Refat, Abu & Al-Imran, Abdullah. (2020). A Convolutional Neural Network for Real-time Face Detection and Emotion & Gender Classification. 15. 37-46.10.9790/2834-1503013746.

[22] Alastair Breeze-https://www.alastairbreeze.com/experiments/machine-learning/facial-keypoint-detection/

[23] Kajal Mishra https://www.pathpartnertech.com/challenges-faced-by-facial-recognition-system/ August 18, 2022

[24] Kaggle competition- https://www.kaggle.com/code/swatisk2702/facial-keypoints-detection-using-inception-model

[25] Danielnouri.org-Using convolutional neural nets to detect facial keypoints tutorial.

[26] Convolutional neural network-https://www.ibm.com/cloud/learn/convolutional-neural-networks

[27] Deep learning network : https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures

[28] Adam Optimizer: https://keras.io/api/optimizers/adam