EasyChair Preprint

№ 7903

Revisiting Facial Key Point Detection - an Efficient Approach Using Deep Neural Networks

Prathima Dileep, Bharath Bolla and Sabeesh Ethiraj

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

May 4, 2022

REVISITING FACIAL KEY POINT DETECTION - AN EFFICIENT APPROACH USING DEEP NEURAL NETWORKS

Abstract Facial landmark detection is a widely researched field of deep learning as this has a wide range of applications in many fields. These key points are distinguishing characteristics points on the face, such as the eyes center, the eye's inner and outer corners, the mouth center, and the nose tip from which human emotions and intent can be explained. The focus of our work has been evaluating transfer learning models such as MobileNetV2 and NasNetMobile, including custom CNN architectures. The objective of the research has been to develop efficient deep learning models in terms of model size, parameters, and inference time and to study the effect of augmentation imputation and fine-tuning on these models. It was found that while augmentation techniques produced lower RMSE scores than imputation techniques, they did not affect the inference time. MobileNetV2 architecture produced the lowest RMSE and inference time. Moreover, our results indicate that manually optimized CNN architectures performed similarly to Auto Keras tuned architecture. However, manually optimized architectures yielded better inference time and training curves.

Keywords: Inference Time, Efficient Transfer Learning, Deep Learning, MobileNetV2, NasNetMobile, Custom CNN, Keras-autotuner

1 Introduction

The face is critical in visual communication. Numerous nonverbal messages, such as human identity, intent, and emotion, can be automatically extracted from the face. Localizations of facial key points are required in computer vision to extract nonverbal cues of facial information automatically. The term "facial appearance" refers to the distinct patterns of pixel intensity around or across facial landmarks or key points. These key points represent those critical features on a human face, such as the eyes, nose, eyebrows, lips, and nose from which information about a person’s emotion or intent can be identified. Once correctly identified, they can be used to train deep learning algorithms to perform various classification tasks. Their applications include computer interaction, entertainment, drowsiness detection, biometrics, emotion detection, security surveillance, and a range of medical applications. However, the practical applications of these models depend on the

speed of inference of these models and their deployability on Edge and mobile devices that have lower computational powers. This research aims to evaluate various transfer learning and custom models in terms of inference time, model size to test their deployability on Edge / mobile devices.

In this work, we used the Facial Key Point Detection dataset from Kaggle. The dataset consists of the training variables and 15 target variables, the facial key points representing various facial features. Deep learning models using custom and transfer learning architectures such as Resnet50, MobileNetV2, NasnetMobile have been built using baseline and also combining various augmentation techniques to identify the ideal model. Additionally, the architectures have been evaluated in terms of parameter count, disc requirements, and inference timings to determine their suitability for deployment on computationally less intensive devices. We have compared our results with other state-of-the-art architectures and found that our models have higher efficiency, hence achieving the objective of this research.

2 Lıterature Review

Facial landmark detection algorithms can be classified into three broad categories [1] based on how they model the facial appearance and shape: holistic, Constrained Local Model (CLM), and regression-based. Holistic methods mainly include Active Appearance Models (AAM)[2] and fitting algorithms. AAM works on the principle of learning from the whole face patch and involves the concept of PCA, wherein learning takes place by calculating the difference "I" between the greyscale image and an instance of the model. The error is reduced by learning the parameters like any conventional machine learning algorithm. CLM methods are slightly better than the holistic approaches as they learn from both the globalized face pattern and the local appearance from the nearby facial keypoints. They can be probabilistic or deterministic. They consist of two steps [3], the initial step where the landmarks are located independent of the other landmarks. In this second step, while updating the parameters, the location of all the landmarks is updated simultaneously. In regression based approaches, there is no initial localization of the landmark; instead, the images are mapped directly to the co-coordinates of these landmarks, and the learning is done directly. These methods may be direct or cascaded. However, with the advent of deep learning algorithms, convolutional neural networks have replaced conventional regression methods with state-of-the-art results. These methods are faster and more efficient. Convolutional neural networks using LeNet have been used in many state-of-the-art works. The principles of LeNet have been

used to build many custom architectures, which have shown reduced training time [4] and reduced RMSE scores.

The performance of a machine learning model also depends mainly on the type of algorithm being used. Some of the popular datasets [1] on which deep learning algorithms have been used with promising results are BU-4DFE with 68 landmark points (RMSE – 5.15), AFLW with 53 landmark points (RMSE-4.26), AFW with five landmark points (RMSE – 8.2), LFPW with 68 landmark points (RMSE – 5.44), Ibug 300-W with 67 landmark points (RMSE – 5.54). Most of the deep learning algorithms have utilized methods such as Task constrained deep convolutional network (TCDCN) [5], Hyperface[6], 3-Dimensional, Dense Face Alignment (3DDFA) [7], Coarse to Fine Auto Encoder Techniques (CFAN) [8] in achieving relatively higher accuracies.

Inception architecture [9] has been used on the similar kaggle dataset achiveing a RMSE score of 2.91. Resnet have also been used in the work done by [10] achieving a RMSE of 2.23. A similar work done using the LeNet architecture [4] achieved a RMSE score of 1.77. As more and more evidences were being produced in favour of custom architectures, focus was then directed on the bulding of Custom CNN networks for facial key point detection. A comparative study was done [11]using both custom and transfer learning architectures. Custom architectures were able to achieve lower RMSE scores(1.97). A similar custom model consisting of 14 layers [12] produced an RMSE score of 1.75. As evident above, achieving higher accuracy with lesser errors by making deep learning algorithms more efficient and precise has been the target of various studies.

Tuning of deep learning models is also critical in achieving high accuracies. The keras tuner library[13] has been widely used to achieve this. Fine-tuning efficiency has been further established in the classification plant leaves disease [14] where fine-tuning architectures such as Resnet50, DenseNet121, InceptionV4 and VGG16 have been used .

Lightweight models such as Mobilenet and NasnetMobile have been gaining popularity recently due to the ease of their deployability. Mobilenet [15] utilizes the concept of depth-wise separable convolution to reduce the number of training parameters without affecting the accuracy of a model. They are ideal for tasks such as recognition of palm prints [16], breast mammogram classification[17], and the identification of proper wearing of facemask[18]. Similar models like Mobilenet have also been built to achieve similar accuracy with fewer parameters, as in the case of PeleeNet[19].

3 Research Methodology

3.1 Dataset description

The dataset for this paper has been taken from the Kaggle competition[20]. There are 7049 images in this dataset and 15 facial key points representing various parts of the face such as eyebrows, eyes, nose, and lips in the training dataset. These facial key points represent the target variables. The test dataset consists of 1783 images. The dataset consists of images of 96x96 size with a one-channel dimension (grayscale images). The distribution of null values is shown below in Figure 1. 69.64% of the data points contain alteast one null value in the facial key points, while 30.36% of the images consists of all key points

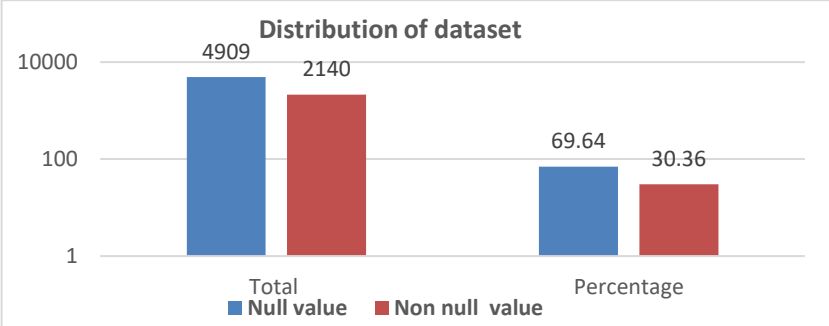

Figure 1. Class Imbalance

3.2 Image Preprocessing

As mentioned below in the models' section, transfer learning architectures such as MobileNetV2 and Nasnet are used on this dataset and custom-designed CNN architectures. These pre-trained networks require the input image to be a three-channel dimensional image and the image size to be 224x224x3 in the case of Nasnet. Hence the images are converted to three-channel images and resized accordingly before model training. The raw image, along with the corresponding facial key points, is shown in Figure 2.

Figure 2. Visualization of Images

3.3 Imputation techniques

Two different types of imputation strategies have been implemented here.

Forward fill & K-Nearest Neighbour (KNN) imputation Forward fill is an imputation technique where the subsequent null values are fileld with the previous valid observations. KNN works on imputing the missing value by predicting the nearest neighbor to a particular datapoint.

3.4 Data Augmentation

Figure 3 depicts the many types of augmentation. The images have been augmented with random rotation, brightness, shift, and noise. These procedures were applied offline on the dataset's non-null subset.

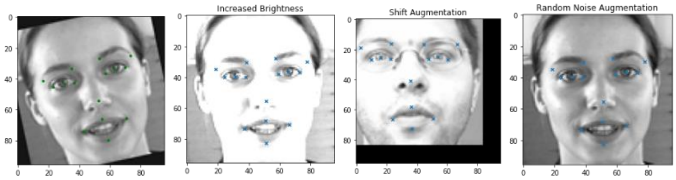

Figure 3. Rotation, Brightness, Shift and Random noise augmentation

3.5 Inference Time

The Inference times of various models have been calculated on 100 images. It can be defined as shown in Equation 1

$$Inf\ time\ \ on\ 100\ images = \frac{Inference\ time\ of\ the\ Total\ test\ dataset}{Number\ of\ test\ images\ in\ the\ test\ dataset}$$

Equation 1. Inference Time Calculation

3.6 Loss functions

The current problem is framed as a regression model where the target variable is a continuous numeric variable, the loss function used here is mean squared error. Mean squared error is defined by the following equation.

$$Mean\ Squared\ Error\ = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

Equation 2. Mean Squared Error - Loss function

3.7 Evaluation metrics

The evaluation metric used in this regression problem is the root mean squared error (RMSE) as shown in Equation 3

$$Root\ Mean\ Squared\ Error\ = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

Equation 3. Root Mean Squared Error

3.8 Model Architecture

Two different models have been built here, Custom models and Transfer learning models, namely MobileNetV2 and NasnetMobile. Tuning is done use Keras tuner library.

Table 1. Parameter / Model size comparison of all architectures

Custom Models	Total parameters	Model size (MB)
Baseline CNN model	1,890,366	7.6
Manually Optimized CNN	235,834	**1.0**
Keras Optimized CNN- No imputation	306,750	1.27
Keras Optimized CNN - Forward fill	246,478	**1.03**
Keras Optimized CNN - KNN imputed	246,062	1.58
Keras Optimized CNN - Augmentation	364,318	1.50
MobilenetV2	2,257,984	9.66
NasNetMobile	4,301,426	18.48

Custom Models. The custom models are tuned sequentially to arrive at the best-performing model in terms of RMSE scores. Three different custom models have

been built using baseline architecture, manual tuning, and Keras auto-tuning. The number of parameters in the model is shown below in Table 1. Complete fine-tuning of transfer learning architectures has also been done. Tuning of the model results in a reduced number of parameters. Manually tuned Custom models have the least parameters with an insignificant difference in RMSE scores, as seen in Figure 7. Further, the tuned model's size is lesser than non-tuned models, with **manually tuned models having the least size** (1.0 MB). The model architecture of the manual tuned and the Keras tuned model is shown below in Figure 4.

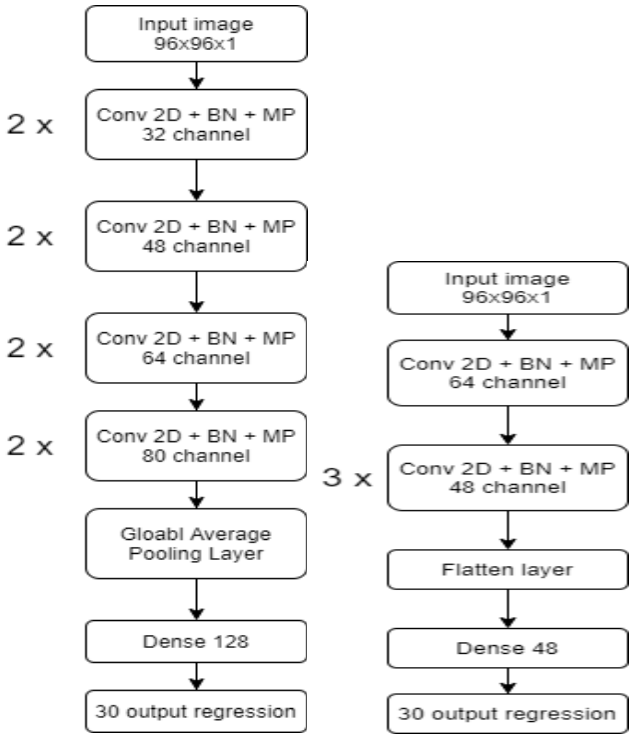

Figure 4. Manually Tuned CNN (Left) and Keras Tuned CNN architecture (Right)

MobileNetV2 and NasNetmobile. Transfer learning architectures such as MobileNetV2 and Nasnetmobile have been customized to solve our regression problem. The original weights from the Imagenet classification have been used. The topmost softmax classification has been replaced with a GAP + Regression (Dense Layer) to predict the facial key points. The models are experimented with using the original baseline weights of imagenet and by completely fine-tuning all the layers of the architecture to evaluate the RMSE scores and inference time on prediction. The model architectures are shown in Figures 5 and 6

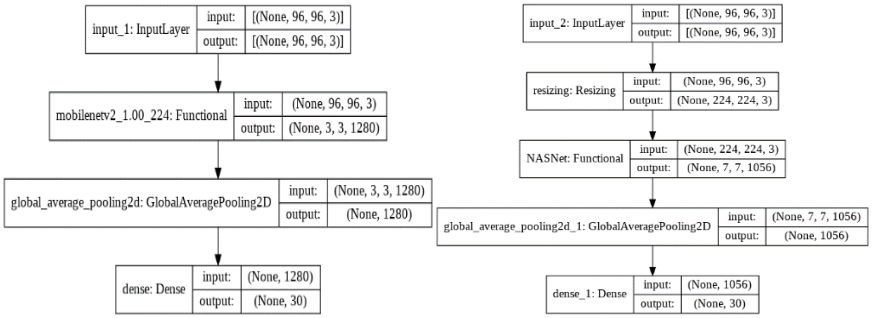

Figure 5. MobilenetV2 (left) and NasnetMobile architecture (right)

4 Results

The results of the experiments have been explained in the following subsections consisting of Evaluation of RMSE scores, model size, and number of parameters

4.1 Evaluation of RMSE scores

RMSE scores on the test dataset have been calculated for both custom and transfer learning models, as shown in Figure 7 and Figure 8.

Huge Parameters of Baseline models. The initial baseline model was created using the conventional architecture without any tuning of the layers among the custom models. Figure 7 show that the Custom baseline model outperformed the manually optimized and Keras fine tuner optimized models; however, manually optimized models performed similarly to Keras fine tuner optimized models.

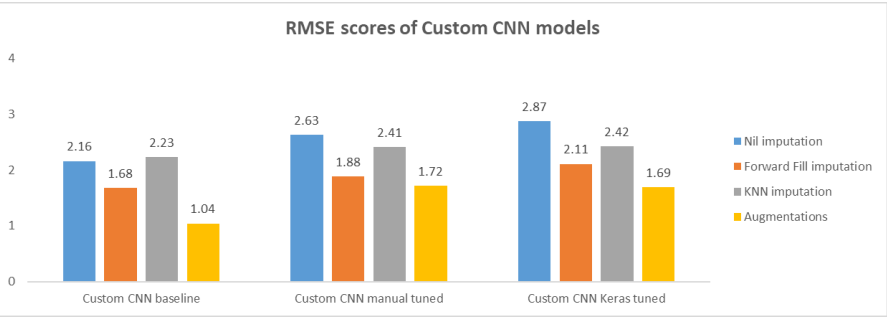

Figure 6. RMSE scores of Custom CNN models

It's worth noting that both fine-tuned Mobilenet and Nasnet trained on augmented data exhibit a 4-5x improvement in RMSE scores compared to their non fine-tuned counterparts (Figure 8). Surprisingly, compared to its non-finetuned counterpart, fine-tuned Mobilenet demonstrated a 2x improvement in RMSE on KNN imputed data.

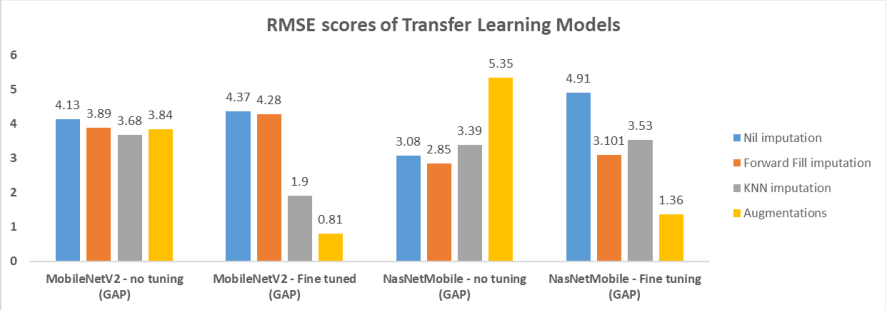

Figure 7. RMSE scores of Transfer Learning Models

Supremacy of models trained on Augmented Data. As seen in the evaluation of custom models in Table 2 and Table 3, that augmentation results in a significant increase in the performance of the models. A sharp decrease in the RMSE scores on the fine-tuned model shows that augmentation performs better than any imputation technique.

Table 2. Comparison of RMSE performance of transfer learning models

Models	No İmputation	Forward fill İmputation	KNN imputation	Aug
MobileNetV2 baseline model	Similar performance	Similar performance	+	+
MobileNetV2 fine tuned			++	+++
NasNet baseline model	Similar performance	Similar performance	Similar performance	+
NasNet Model fine-tuned				+++

4.2 Evaluation of Model size and parameters

Among all the models built, manually tuned custom models have the least number of parameters (235K) and least model size against Keras autotuned custom models that are trained on different kinds of imputation techniques and augmentation. However, in the case of augmentation, Keras autotuned models slightly outperform custom models at the cost of increasing the number of parameters and model size.

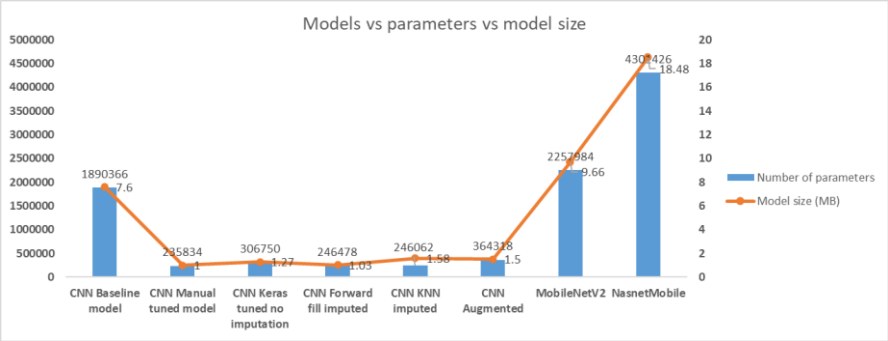

Figure 8. Model parameters vs. Model Size – All Models

4.3 Inference time analysis

The ultimate performance depends on the speed at which an inference can be made on the test dataset with the least computational requirements. Table 4 shows the inference time on 100 images by various models on a Colab CPU.

Table 3. Inference Time Analysis

Model	No impute (sec)	Forward Fill (sec)	KNN Impute (sec)	Aug(sec)
CNN Baseline model	1.99	1.97	1.98	2.01
CNN Manual tuned model	**1.4**	**1.33**	**1.34**	**1.34**
CNN Keras tuned	2.72	1.52	4.19	3.58
MobileNetV2 - Baseline	**0.89**	**0.86**	**1**	**0.83**
MobileNetV2 - Fine tuned	**0.84**	**0.82**	**0.82**	**0.88**
NasNetMobile - Baseline	8.46	8.4	8.17	7.87
NasNetMobile - Fine tuned	7.95	7.96	8.01	7.68

Architectural efficiency in Inference Time. The inference time of a model depends on both the number of parameters and the architecture. Among all models, MobileNetV2 has the quickest inference. The enormous training parameters (twice that of MobileNetV2) account for NasNetmobile's long inference times. Manually tuned models come in second. In contrast to custom CNN models, MobileNet has ten times the number of parameters and works two times faster. Augmentation does not affect the inference time in a regression scenario, as seen from the analysis

4.4 Evaluation of Training Curves

Training curves for various models are shown below to identify the best performings model in this scenario.

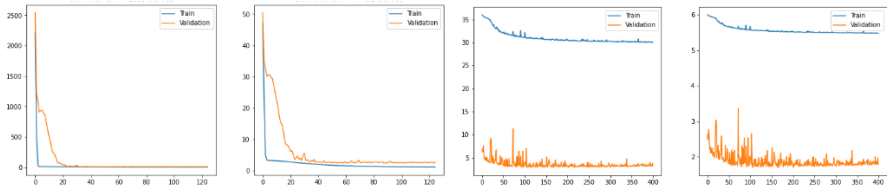

Figure 9. Custom CNN Manual Tuned (Left) Vs Custom CNN Keras Tuned (Right)

Manual Tuning vs. Keras autotuning. Manually tuned models exhibit more reliable model fitting training curves than Keras auto tuned models, as illustrated in Figure 10.

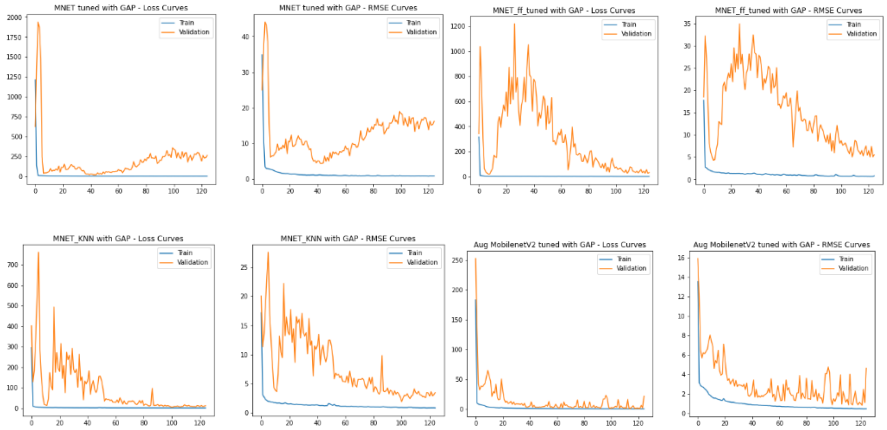

Figure 10. MobileNetV2 - No impute, Forward Fill, KNN impute, Augmentation (Top to bottom)

MobileNetV2 vs NasnetMobile. Figures 11 and 12 show that Nasnet mobile has better training curves than Mobile Net architecture for all imputation techniques and augmentation. The better training curves may be attributed to higher parameters of Nasnet. However, when considering inference times, RMSE scores, and parameter counts, MobileNetV2 outperforms Nasnet.

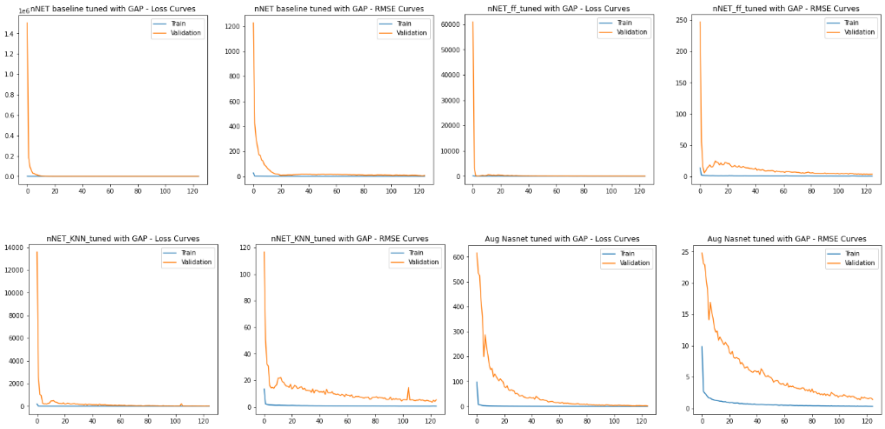

Figure 11. NasnsetMobile - No impute, Forward Fill, KNN impute, Augmentation (Top to Bottom)

4.5 Visualization of Test Images

Figure 13 shows various augmented models' predictions of facial key points. Varying performances by different models are observed in the images below. However, the images only represent a sample of the total test dataset, and hence no meaningful conclusion can be drawn.

Figure 12. Facial Key Point Predictions by CNN manual tuned/Keras autotuned, MobilenetV2 and Nasnet on Augmentation

5 Conclusion

In this work, we conducted experiments on the facial key point detection dataset by building custom CNN models optimized manually and using Keras fine-tuner. Further transfer learning architectures, non-finetuned and fine-tuned MobileNetV2 and NasNetMobile were used as baselines to evaluate custom-built CNN architecture. In addition, we compared the effectiveness of imputation and augmentation. The following are the conclusions of our work which can be summarized below.

- Manually optimized custom CNN models outperform or are comparable to auto-tuned Keras optimized models. On the other hand, manually tuned custom CNN models may be ideal when considering training curves, model size, and model parameters.

- MobileNetV2 outperforms all other models with the fastest inference times but slightly compromising the model size and parameters.

- In both custom CNN and transfer learning models, augmented models have lower RMSE scores, proving that augmentation is superior to imputation.

- Furthermore, there is no significant difference in performance between baseline non-tuned and baseline completely fine-tuned models, demonstrating that transfer learning models must be fine-tuned selectively in terms of the number of layers for a given dataset.

- The experiments demonstrate that architectural efficiency significantly impacts model performance and inference time, as demonstrated by the MobileNetV2 architecture, which uses depth-wise separable convolutions.

- Moreover, our models have the lowest RMSE compared to other **state-of-the-art architectures** ([4], [10], [11], [12]), and to our knowledge, this is one of the very few studies that evaluated models on size, inference time, parameters and RMSE

6 References

[1] Y. Wu and Q. Ji, “Facial Landmark Detection: A Literature Survey,” *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, Feb. 2019, doi: 10.1007/s11263-018-1097-z.

[2] T. F. Cooles, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001, doi: 10.1109/34.927467.

[3] A. Zadeh, Y. C. Lim, T. Baltrušaitis, and L.-P. Morency, “Convolutional Experts Constrained Local Model for 3D Facial Landmark Detection.”

[4] N. Agarwal, A. Krohn-Grimberghe, and R. Vyas, “Facial Key Points Detection using Deep Convolutional Neural Network - NaimishNet,” pp. 1–7, 2017, [Online]. Available: http://arxiv.org/abs/1710.00977

[5] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “Facial Landmark Detection by Deep Multi-task Learning.”

[6] R. Ranjan, V. M. Patel, and R. Chellappa, “HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition,” Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.01249

[7] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, “Face Alignment in Full Pose Range: A 3D Total Solution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 78–92, Jan. 2019, doi: 10.1109/TPAMI.2017.2778152.

[8] J. Zhang, S. Shan, M. Kan, and X. Chen, “Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment.”

[9] C. Mao, “Facial Keypoints Detection with Inception Structure,” pp. 3–5, 2016.

[10] S. Wu, J. Xu, S. Zhu, and H. Guo, “A Deep Residual convolutional neural network for facial keypoint detection with missing labels,” *Signal Processing*, vol. 144, pp. 384–391, Mar. 2018, doi: 10.1016/j.sigpro.2017.11.003.

[11] S. Shi, “Facial Keypoints Detection,” pp. 1–28, 2017, [Online]. Available: http://arxiv.org/abs/1710.05279

[12] R. Gao, “Facial Keypoints Detection with Deep Learning,” *Journal of Computers*, vol. 13, no. 12, pp. 1403–1410, 2018, doi: 10.17706/jcp.13.12.1403-1410.

[13] “Introduction to the Keras Tuner | TensorFlow Core.” https://www.tensorflow.org/tutorials/keras/keras_tuner (accessed Nov. 24, 2020).

[14] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[15] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, [Online]. Available: http://arxiv.org/abs/1704.04861

[16] A. Michele, V. Colin, and D. D. Santika, “Mobilenet convolutional neural networks and support vector machines for palmprint recognition,” in *Procedia Computer Science*, 2019, vol. 157, pp. 110–117. doi: 10.1016/j.procs.2019.08.147.

[17] “Transfer Learning in Breast Mammogram Abnormalities Classification With Mobilenet and Nasnet.”

[18] B. Qin and D. Li, “Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19,” *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–23, Sep. 2020, doi: 10.3390/s20185236.

[19] R. J. Wang, X. Li, and C. X. Ling, “Pelee : A Real-Time Object Detection System on Mobile Devices,” no. NeurIPS, pp. 1–10, 2018.

[20] “Facial Keypoints Detection | Kaggle.” https://www.kaggle.com/c/facial-keypoints-detection/data (accessed Jun. 28, 2020).