Gender Classification using CNN model

Đặng Hải Bình, Võ Thành Nam, Nguyễn Hữu Đức

Note: Trường hợp bài bị hỏng format, vào link đính kèm: https://docs.google.com/document/d/1Cf64DwJAJ8JqPJ2XMqtEZGy6vhX6tjYilTXPMb9lz9c/edit#

Abstract

A proposal for solving the gender classification problem using convolutional neural networks (CNN). The model proposed shows its simplicity without losing its high accuracy when handling data. Our model consists of a base model with 5 extra convo layers and 4 dense layers, plus 202601 with 64x64 pixel RGB images, and after 8 epoch runs was able to give out the accuracy of 95% with the VGG16 model modification. This result shows the superiority of the current model with others, confirming that the proposed CNN is an effective solution for gender recognition.

1. Introduction:

Gender classification, which is a philosophical problem that our kind has had since ancient times. As we use our skill of recognition to check for key features on the human faces, research has shown that we can do the same with machines. As those are able to recognize key elements, they can also be used as the backbone of many security systems, improving performances of face recognition applications in biometrics, human–computer interactions, and computer vision.

Now we face a new problem if we want to use machine learning for gender classification - as facial features differ from person to person, meaning if we are able to pick all the features for a machine to learn, it will mostly come out be overfitted. Instead, deep learning is the way this

problem should be handled. As the data can be analyzed by the machine, it will be able to learn to pick hidden key features and use them to classify the gender.

As we chose to go with the deep learning path, we also want to point out the reason for using CNN [1] . Its built-in convolutional layer reduces the high dimensionality of images without losing its information, which makes it much simpler to handle many parameters, and shows that it is the perfect candidate for solving this problem.

2. Related Work:

For our project, gender classification using image processing, there are quite a lot of articles that work on this. Most of the article was written in recent times, ranging from 2012 to now. But for newer and better model usage and efficiency, we will only use and research articles that were written after 2015.

2.1. Age and Gender Classification using Convolutional Neural Networks (2015):

[2] In this paper, they demonstrate that performance on these tasks can be significantly improved by learning representations using deep convolutional neural networks (CNN). In order to do this, they suggest a straightforward convolutional net architecture that may be applied even with a finite supply of training data. Obtaining a sizable, manually annotated image training set for age and gender estimates from social image archives involves either access to the individuals' often-private personal information (their birth date and gender) or is laborious and time-consuming.

Only three convolutional layers and two fully linked layers with few neurons make up the network. After applying 96 filters with a size of 377 pixels to the input in the first convolutional layer, The second convolutional layer, which has 256 filters with a size of 96 x 5 x 5, processes the 96 x 28 x 28 output of the preceding layer. The third and final convolutional layer applies a set of 384 filters with a size of 256 x 3 x 3 pixels to the 256 x 14 x 14 blob, followed by a rectified linear operator (ReLU), a max pooling layer that takes the maximum value of 3 x 3

regions with two-pixel strides, and a local response normalization layer are then defined as the three subsequent convolutional layers. A ReLU and a dropout layer are placed after a first fully connected layer that receives the output of the third convolutional layer and has 512 neurons, a second fully connected layer that receives the 512-dimensional output of the first fully connected layer, and 512 neurons again. a third, completely interconnected layer that corresponds to the final age or gender classes. As a result, the accuracy for age and gender is 84.7% and 86.8%, respectively.

2.2. Gender classification: a convolutional neural network approach (2016):

[3] Comparing the proposed CNN architecture to other CNN solutions used in pattern recognition, they find that the design complexity is significantly lower. By combining the convolutional and subsampling layers, the CNN is reduced to only four processing layers. Contrary to conventional CNNs, they substitute cross-correlation for the convolution operation to lessen the computational load. A second-order backpropagation learning algorithm with annealed global learning rates is used to train the network.

The proposed CNN solution's performance is assessed using two publicly accessible face databases from SUMS and AT&T. On the SUMS and AT&T databases, they achieve classification accuracy levels of 98.75% and 99.38%, respectively. A 32 x 32 pixel face image can be processed and classified by the neural network in less than 0.27 milliseconds, which translates to a very high throughput of more than 3700 images per second. Within 20 epochs or less, training converges.

2.3. Convolutional Neural Networks for Age and Gender Classification (2016):

[4] In this paper, they create a benchmark for the task based on cutting-edge network architectures and demonstrate how chaining the age and gender predictions can increase overall accuracy. An image of a human face that is 256x256 in size and then cropped to 227x227 is the input to their algorithm.

There are 3 convolution layers, followed by 3 fully connected layers, With stride 4 and padding 0, Conv1-96 filters of size 3x7x7 are convolved to produce an output volume that is 96x56x56 in size. Stride 1 and padding 2 are convolved with Conv2-256 filters of size 96x5x5 to produce an output volume size of 256x28x28. Conv3-256 filters are convolved with stride 1 and padding 1 in a 256x3x3 matrix. This is followed by a local-response normalization (LRN), a max pooling pooling to reduce the size, and a ReLU.

Following a ReLU layer and dropout layer, FC6 has 512 neurons fully connected to the 256x7x7 output of Conv3, and FC7 has 512 neurons fully connected to the 1x512 output of FC6. The un-normalized class scores for either gender or age are produced by FC8- 2 or 8 neurons completely connected to the 1x512 output of FC7, respectively. The results for chained net accuracy are summarized as follows: 84,1%. As suggested, this chained structure improves classification accuracy compared to the conventional method of training age classifiers on both genders simultaneously.

2.4. Local Deep Neural Networks for Age and Gender Classification (2017):

[5] Gender recognition using local deep neural networks has lately been developed. Despite having excellent efficiency, they require a lot of computational power to train. In this study, they introduce a local deep neural network that greatly shortens training time through a simplified design.

Images are initially transformed to grayscale (from 0.0 - black to 1.0 - white).The next step is to create the nine 30-by-30 patches. Pixel values were adjusted to the standard Gaussian distribution with zero mean and unity variance for each and every patch. They recommend using 9 overlapping patches per image, which cover the entire face area, as opposed to the original method's suggestion of using hundreds of patches per image.

Due to the fact that only 9 patches rather than hundreds are extracted from each image, a significant reduction in training time is achieved at the cost of a marginal decline in performance. For the task of gender and age classification, they put the proposed modified local deep neural networks approach to the test on the LFW and Adience databases. The performance is up to 1% worse than the algorithm's original version for both tasks and both databases.

2.5. Deeper understanding on CNN model:

The field of Deep Learning has materialized a lot over the past few decades due to efficiently tackling massive datasets and making computer systems capable enough to solve computational problems. Hidden layers have ushered in a new era, with the old techniques being non-efficient, when it comes to image processing-based problems.

The most common aspect of any A.I. models that require a massive amount of data to train. Yann LeCun was the first to introduce convolutional neural networks [1] . Convolutional Neural uses a very special kind of method which is known as Convolution. The mathematical definition of convolution is a mathematical operation being applied on the two functions that give output in the form of a third function that shows how the shape of one function is being influenced, modified by the other function.

The Convolutional neural networks(CNN) consists of various layers of artificial neurons. Artificial neurons, similar to those neuron cells that are being used by the human brain for passing various sensory input signals and other responses. When we give an input image into a CNN, each of its inner layers generates various activation maps. The output of the first layer is being fed as an input of the next layer, which in turn will extract other complex features of the input image like corners and combinations of edges.

3. Data preparation:

The dataset which our team uses in this project is the: Gender Dataset - Yasir Hussein Shakir [6] . The dataset is divided into 2 sub datasets: female and male. The dataset consists of around: 120000 for female and 85000 for male (total: 202601 image).

Male	Female

Figure 1: Some example images of dataset

Dataset	Train	Validation	Test	Total
Female	92845	13778	11542	118165
Male	67155	8820	8459	84434

Table 1: Dataset divided structural

The data have been cleaned and resized to 178x218, compiled of human faces in RGB format. All of our models train on the same dataset.

4. Method:

We use some pretrain object detection model on the Imagenet dataset as base, and add more layers to it. We add 5 convolution layers with 3x3 kernel size, and n filters respectedly: 512,128,384,384,500. With fully connected layers we have 2 dense layers with 2048 nodes and 1 output layer with 1 node. Our model will be retrained with: learning rate = 0.001, epochs = 8 and 256 steps per epoch. After retraining the model using sigmoid activation function as the output we have the probability to determine if the picture is male or female(female<0.5<male).

Input data will be scaled to a 64x64 numpy array and normalized.

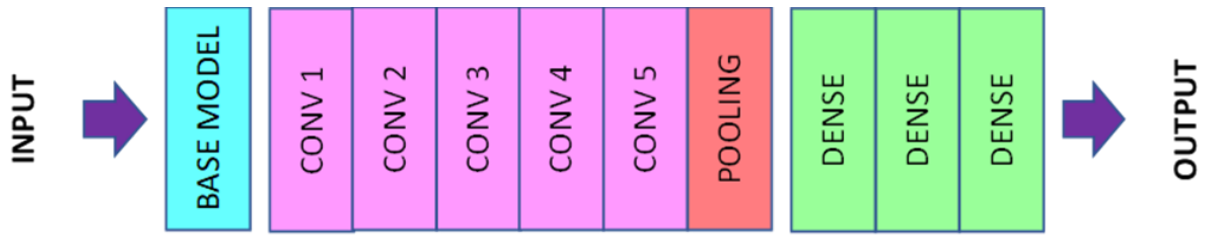

Figure 2: Our model architecture

4.1. What is VGG?

VGG [7] , short for Visual Geometry Group, is a commonly used Convolutional Neural Network (CNN) architecture that has several layers. VGGNet is a Convolutional Neural Network architecture proposed by Karen Simonyan and Andrew Zisserman from the

University of Oxford in 2014. The term "deep" refers to the number of layers, with VGG-16 and VGG-19 having 16 and 19 convolutional layers, respectively. VGG is responsible for creating groundbreaking models for object recognition and has become the foundation for many image recognition architectures. VGGNet, developed as a deep neural network, has surpassed baselines on various datasets and tasks beyond ImageNet and remains one of the most popular image recognition architectures in use today. The input to VGG based convNet is a 224*224 RGB image. Preprocessing layer takes the RGB image with pixel values in the range of 0–255 and subtracts the mean image values which is calculated over the entire ImageNet training set.

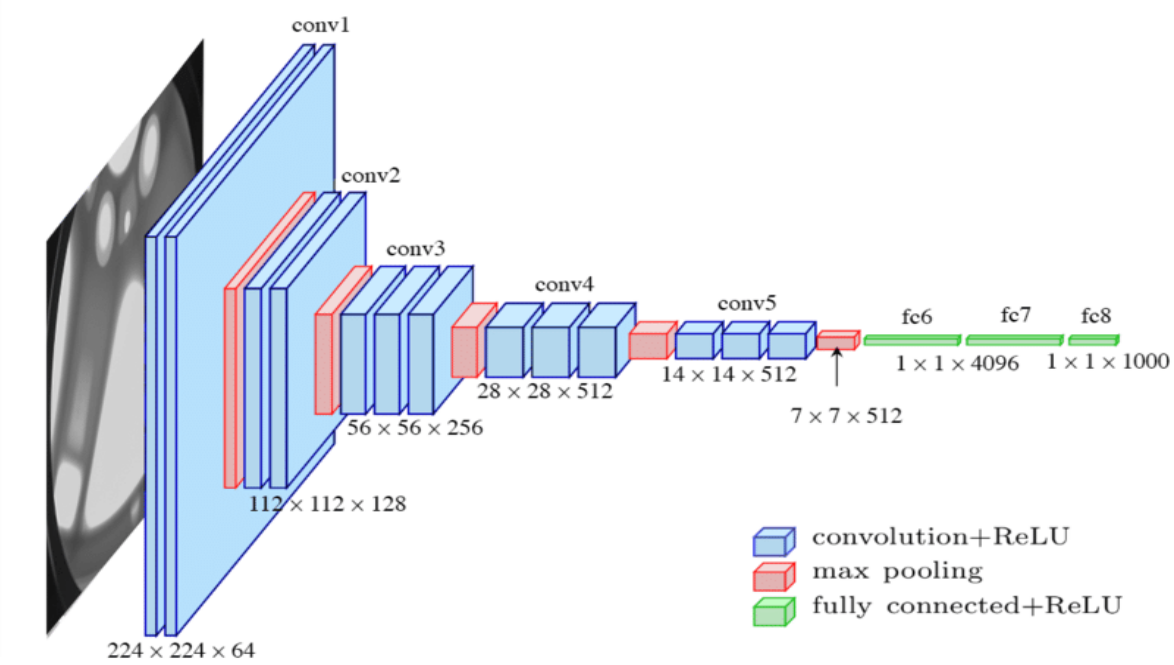

Figure 3: VGG architecture

The input images after preprocessing are passed through these weight layers. The training images are passed through a stack of convolution layers. There are a total of 13 convolutional layers and 3 fully connected layers in VGG16 architecture.The VGG16 model achieves almost 92.7% top-5 test accuracy in ImageNet. ImageNet is a dataset consisting of more than 14 million images belonging to nearly 1000 classes. Moreover, it was one of the most popular models submitted to ILSVRC-2014. It replaces the large kernel-sized filters with several 3×3

kernel-sized filters one after the other, thereby making significant improvements over AlexNet. It has ended up having the same effective receptive field as if you only have one 7 x 7 convolutional layer. The VGG16 model was trained using Nvidia Titan Black GPUs for multiple weeks.

Another variation of VGGNet has 19 weight layers consisting of 16 convolutional layers with 3 fully connected layers and the same 5 pooling layers. In both variations of VGGNet there consists of two Fully Connected layers with 4096 channels each which is followed by another fully connected layer with 1000 channels to predict 1000 labels. Last fully connected layer uses the softmax layer for classification purposes.

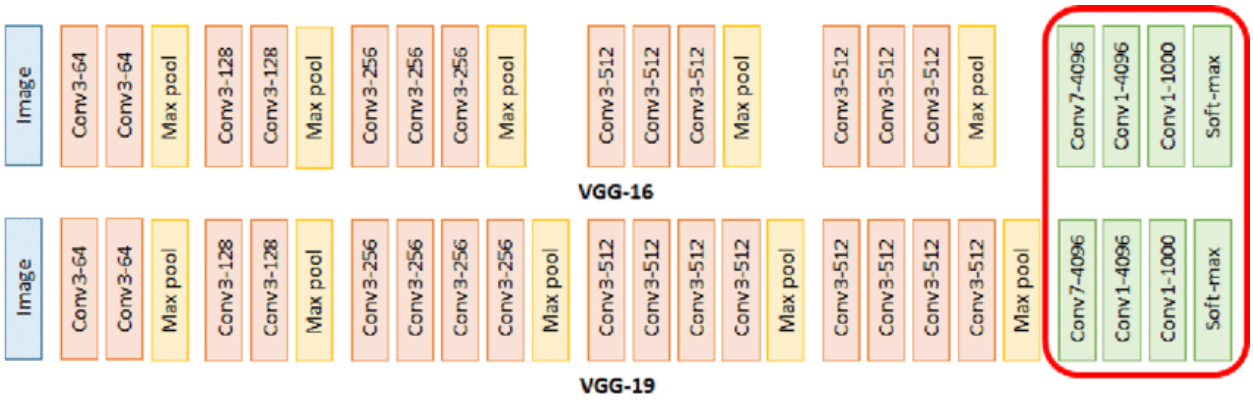

Figure 4: VGG16 vs VGG19 architecture

4.2. What is MobileNet?

MobileNet [8] uses a Convolutional Neural Network (CNN) architecture model to classify images. It is open-sourced by Google. MobileNet architecture is special because it uses very less computation power to run. This makes it a perfect fit for mobile devices, embedded systems, and computers to run without GPUs. MobilenetV1 is the first version of the Mobilenet models. It has more complex convolution layers and parameters when compared to MobilenetV2. MobilenetV2 is the second version of the Mobilenet models. It significantly has a lower number of parameters in the deep neural network. This results in more lightweight deep neural networks. Being lightweight, it is best suited for embedded systems and mobile

devices. MobilenetV2 is a refined version of MobilenetV1. This makes it even more efficient and powerful. The MobileNetV2 models are faster due to the reduced model size and complexity.MobilenetV2 is a pre-trained model for image classification. Pre-trained models are deep neural networks that are trained using a large images dataset. Using the pre-trained models, the developers need not build or train the neural network from scratch, thereby saving time for development.

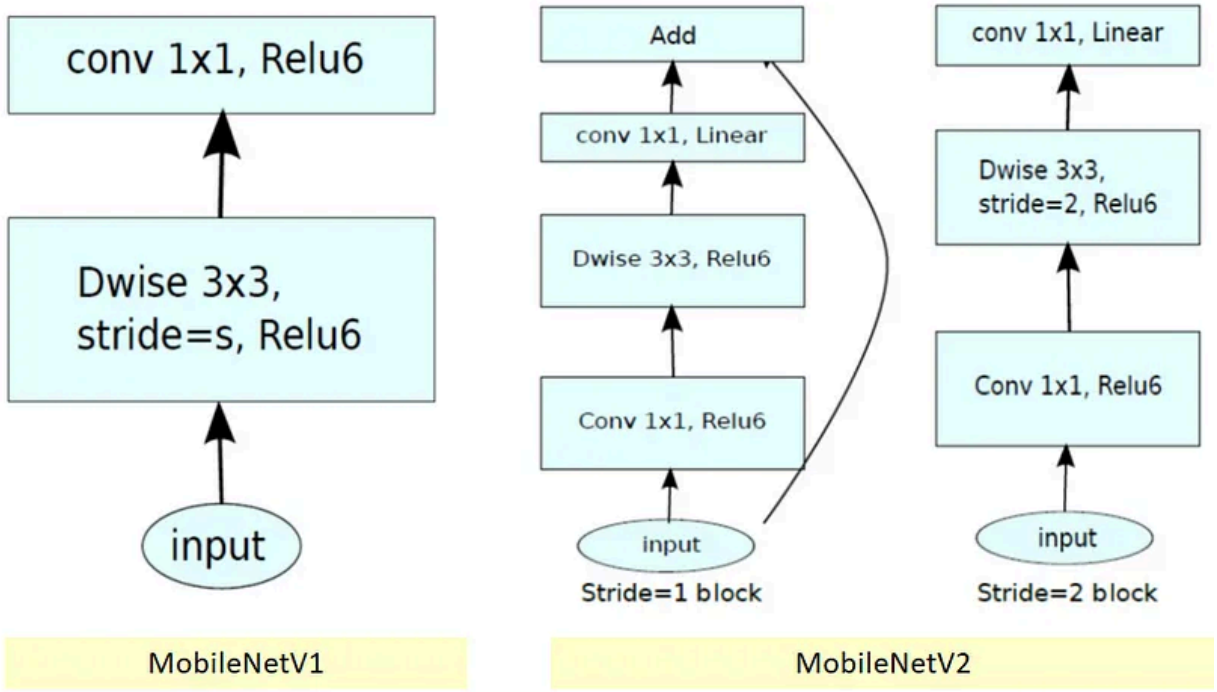

Figure 5: MobileNetV1 vs MobileNetV2 architecture

4.3. What is EfficientNetV2?

The EfficientNet [9] models are designed using neural architecture search. The first neural architecture search was proposed in the paper in 2016. The idea is to use a controller (a network such as an RNN) and sample network architectures from a search space with probability ‘p’. This architecture is then evaluated by first training the network, and then validating it on a test set to get the accuracy ‘R’. The gradient of ‘p’ is calculated and scaled by the accuracy ‘R’. The result (reward) is fed to the controller RNN. The controller acts as the agent, the training and testing of the network act as the environment, and the result acts as the

reward. This is the common Reinforcement learning (RL) loop. This loop runs multiple times till the controller finds the network architecture which gives a high reward (high test accuracy).

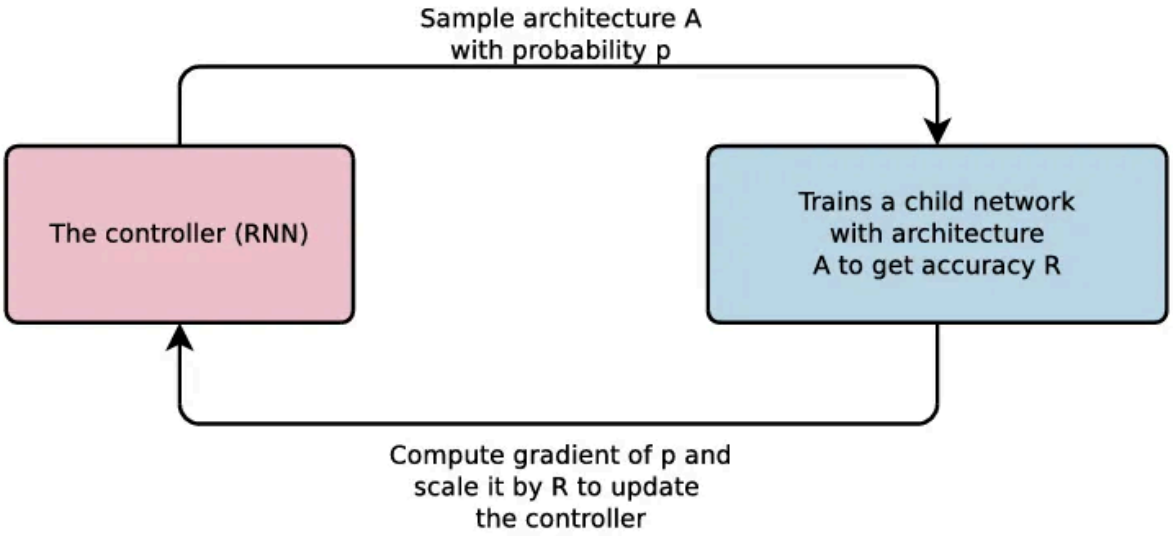

Figure 6: An overview of Neural Architecture Search

The controller RNN samples various network architecture parameters — such as the number of filters, filter height, filter width, stride height, and stride width for each layer. These parameters can be different for each layer of the network. Finally, the network with the highest reward is chosen as the final network architecture.

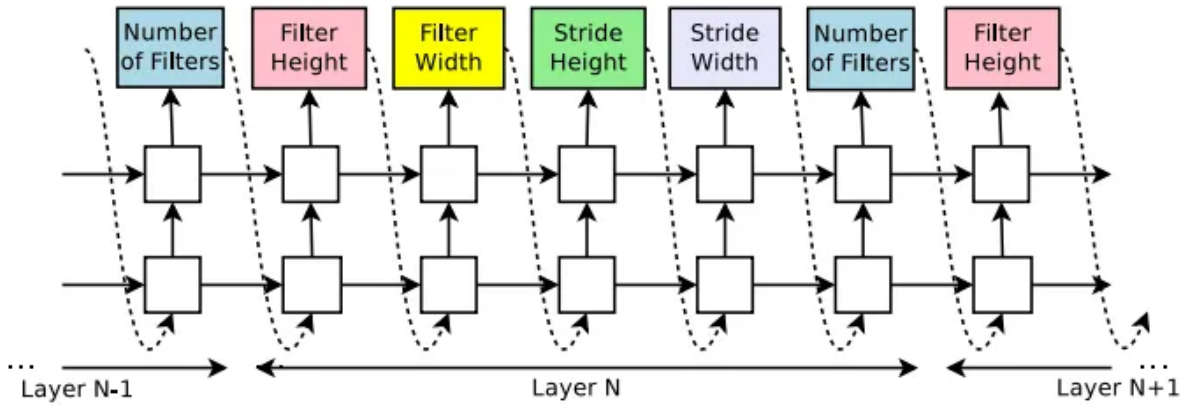

Figure 7: All the different parameters that the controller searched for in each layer of the network

Even though this method worked well, one of the problems with this method was that this required a huge amount of computing power as well as time.

To overcome this problem, in 2017, a new method was suggested. Looked into previously famous Convolutional Neural Network (CNN) architectures such as VGG or ResNet, and figured, that these architectures do not have different parameters in each layer, but rather have a block with multiple convolutional and pooling layers, and throughout the network architecture, these blocks are used multiple times. The authors used this idea to find such blocks using the RL controller and just repeated these blocks N times to create the scalable NASNet architecture. One more very important parameter was considered while deciding the reward, which went into the controller, and that was ‘latency’. So for MnasNet, the authors considered both the accuracy and latency to find the best model architecture.

Finally, the EfficientNet architecture was proposed in 2020. The workflow for finding the EfficientNet architecture was very similar to the MnasNet, but instead of considering ‘latency’ as a reward parameter, ‘FLOPs (floating point operations per second)’ were considered. This criteria search gave the authors a base model, which they called EfficientNetB0. The base EfficientNet-B0 network is based on the inverted bottleneck residual blocks of MobileNetV2, in addition to squeeze-and-excitation blocks. Next, they scaled up the base models' depth, width, and image resolution (using grid search) to create 6 more models, from EfficientNetB1 to EfficientNetB7.

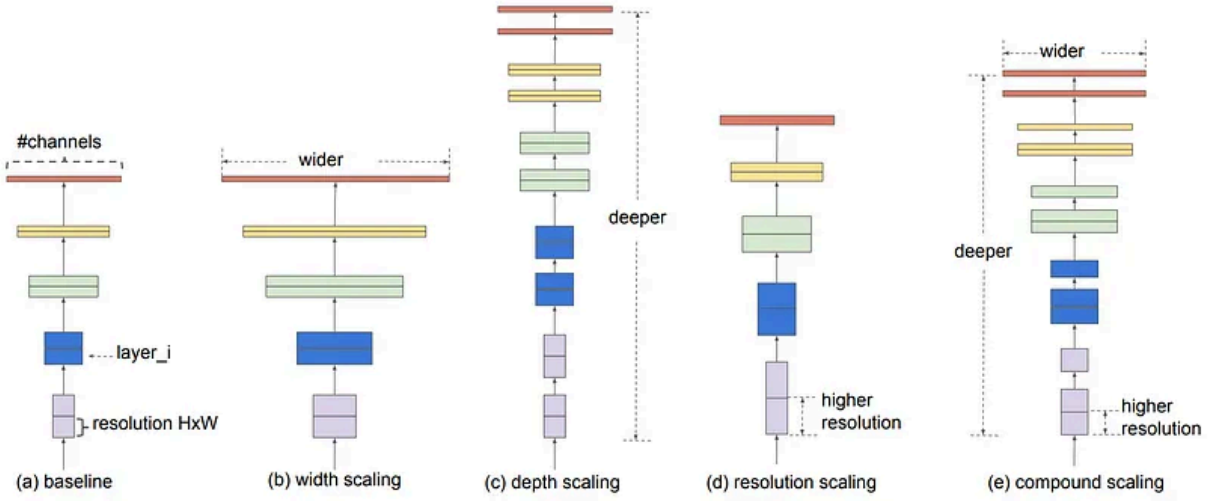

Figure 8: Scaling the depth, width, and image resolution to create different variations of the EfficientNet model

EfficientNetV2 goes one step further than EfficientNet to increase training speed and parameter efficiency. This network is generated by using a combination of scaling (width, depth, resolution) and neural architecture search. The main goal is to optimize training speed and parameter efficiency. Also, this time the search space also included new convolutional blocks such as Fused-MBConv. In the end, the authors obtained the EfficientNetV2 architecture which is much faster than previous and newer state-of-the-art models and is much smaller (up to 6.8x times).

Training time (TPU days)

(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Figure 9: Training and Parameter efficiency of the EfficientNetV2 model compared with other state-of-the-art models

5. Result and discussion:

With our dataset the baseline for accuracy is 58 percent, with all our models exceeding that. To mention, we have to stop some of our models for better precision, with *VGG19* having to stop training on epoch 5, *MobileNetV2* at epoch 7 and *EfficientNetV2* at epoch 5 as well. Our highest accuracy model came from *VGG16* at 95 percent, with more than 30 million parameters, which is the second lowest amount of parameters used, meaning that this model is also lightweight but gives out an incredible efficiency. But we can still use *MobileNetV2* if we want a lighter model as a trade off for a small amount of efficiency. Other models' information is also inform in the table below:

Model	Accuracy	Parameters	Time train(sec)	Loss	Model size(Mb)
MobileNetV2	94%	21,673,861	5128	0.31	254
VGG16	95%	30,591,621	2235	0.13	358
VGG19	91%	35,901,317	1935	0.31	421
EfficientNetV2	88%	72,566,265	3041	0.27	851

Table 2: Performance comparison with each models

It's also worth pointing out that the EfficientNet V2 is the model that uses the most parameters but is unable to give out an accuracy above 90 percent.

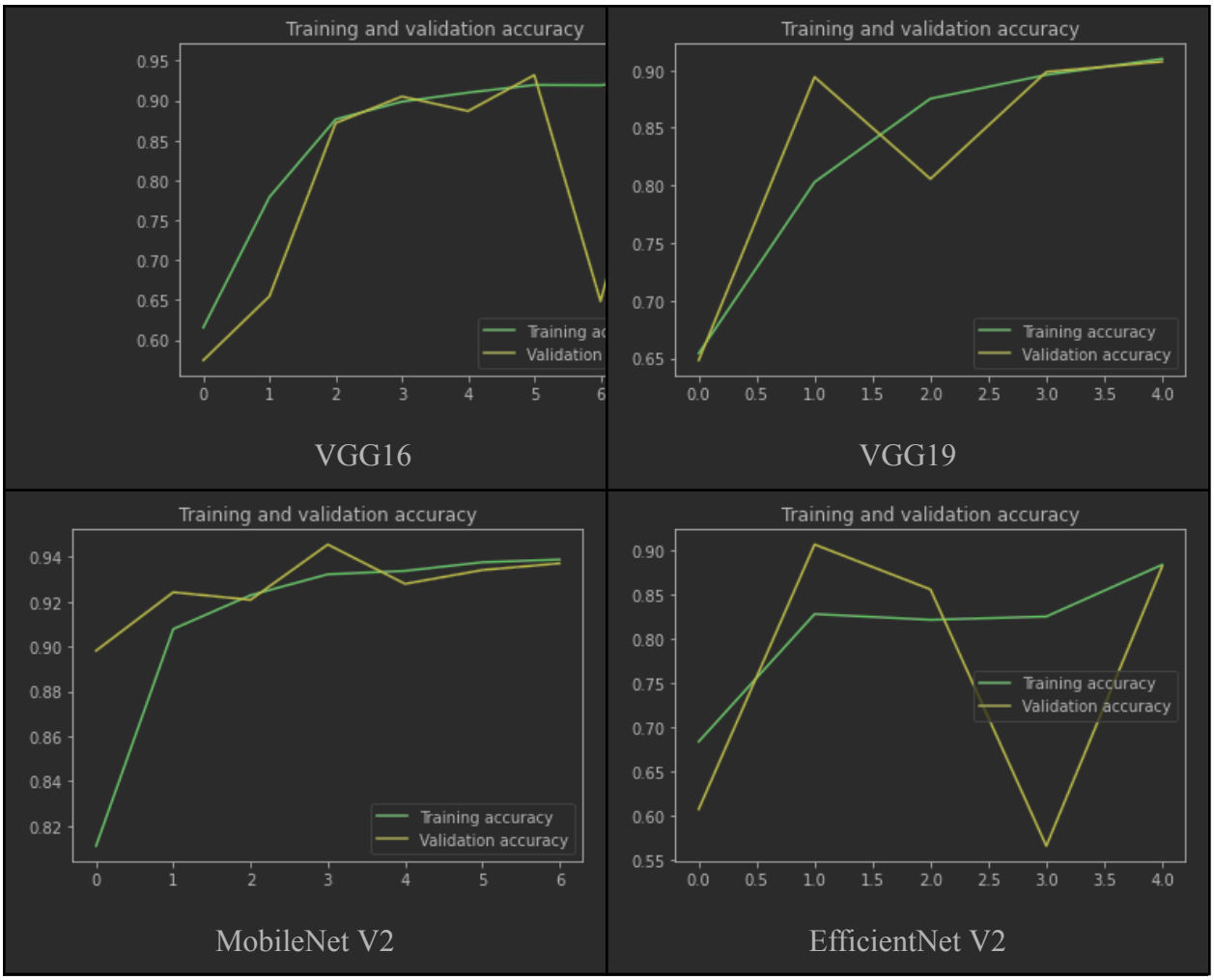

Figure 10: Accuracy of train and validation with each model though epochs

We can see that simple and less parameters models tend to do better than complex ones. With too many parameters, other models are prone to overfitting and lead to the decrease of accuracy. With the VGG16 model, testing shows a balanced result between 2 classes, with good precision and recall so our model is mostly optimized.

Figure 11: Confusion matrix on testset of final VGG16 model

Conclusion:

To conclude our project, we are able to state that VGG16 is the best choice for solving this problem, as it only needs a small amount of parameters but still able to give out a high accuracy score with a moderate training time. This shows we can use VGG16 for many purposes related to image classification in general and gender classification to be more precise. Currently many of our devices already have this AI model for facial analysis, which are CCTV, biometric lock, etc, and more to come in the future as it is still being developed.

Reference:

[1] Sharma, P. (2022, March 1). *Basic Introduction to Convolutional Neural Network in Deep Learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/

[2] Levi, G., & Hassncer, T. (2015). Age and gender classification using convolutional neural networks. *Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvprw.2015.7301352

[3] Liew, S. S., Khalil-Hani, M., Radzi, S. A., & Bakhteri, R. (2016). Gender classification: a convolutional neural network approach. *Turkish Journal of Electrical Engineering and Computer Sciences*, *24*, 1248–1264. https://doi.org/10.3906/elk-1311-58

[4] Convolutional Neural Networks for Age and Gender Classification (2016), by Ari Ekmekji: http://vision.stanford.edu/teaching/cs231n/reports/2016/pdfs/003_Report.pdf

[5] Liao, Z., Petridis, S., & Pantic, M. (2017). Local Deep Neural Networks for Age and Gender Classification. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1703.08497

[6] Y. (2021, September 2). *Gender Classification Using VGG16+CNN*. Kaggle. https://www.kaggle.com/code/yasserhessein/gender-classification-using-vgg16-cnn

[7] Nepal, P. (2021, December 15). *VGGNet Architecture Explained - Analytics Vidhya - Medium*. Medium. https://medium.com/analytics-vidhya/vggnet-architecture-explained-e5c7318aa5b6

[8] Tsang, S. (2021, December 10). *Review: MobileNetV2 — Light Weight Model (Image Classification)*. Medium. https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c

[9] Sarkar, A. (2022, October 8). *EfficientNetV2: Faster, Smaller, and Higher Accuracy than Vision Transformers*. Medium. https://towardsdatascience.com/efficientnetv2-faster-smaller-and-higher-accuracy-than-vision-transformers-98e23587bf04