Building a CNN Model for Gender Classification

Ankur Saha
Department of CSE
American International University-Bangladesh
Dhaka, Bangladesh
18-37913-2@student.aiub.edu

Abstract: This paper is based on a modified convolutional neural network which was implemented and ran on a specific dataset, which differentiates between male and female. After implementing the dataset with predefined CNN architecture like ResNet50 and VGGNet (16 layers), accuracy and loss rate was recorded and compared, which guided a lot in making a modified version of convolutional neural network architecture, so that the model could reach closest to the current achieved accuracy or even beat it. At the end, a visualization was performed to show how the machine is detecting different image patterns with the help of filters. A filter learning process was shown through some images of random convolutional layers and interpretation of the pattern learning was also done.

Keywords: Gender Classification, CNN, ResNet, VGG16, Face Recognition

1. Introduction

In the domains of AI, machine learning, and deep learning, neural networks mimic the function of the human brain, allowing computer programs to spot patterns and solve common problems. It is like a computer system made up of linked nodes that function similarly to neurons in the brain. They can discover hidden patterns and correlations in raw data using various types of algorithms, cluster and categorize it, and learn and improve over time by training. One commonly used neural network, CNN (Convolution Neural Network) is known for investigation of visual images. Multilayer perceptrons are regularized variants of CNNs. Multilayer perceptrons are typically completely connected networks, meaning that each neuron in one layer is linked to all neurons in the following layer. These networks' "complete connectedness" makes them vulnerable to data overfitting. Regularization, or preventing overfitting, can be accomplished in a variety of methods, including punishing parameters during training (such as weight loss) or reducing connectivity (skipped connections, dropout, etc.) CNNs use a different approach to regularization: they take advantage of the hierarchical pattern in data and use smaller and simpler patterns imprinted in their filters to construct patterns of increasing complexity. There are variations of datasets available for image processing to train the computer which then be able to predict the similar images by partitioning them into categories. In this project, the dataset was selected from Kaggle which represents two categories, male and female. It was based on CELEBA aligned dataset where the owner went through and separated the images visually into 1747 female and 1747 male training images. There was also 100 male and 100 female test images with 100 male, 100 female validation images.

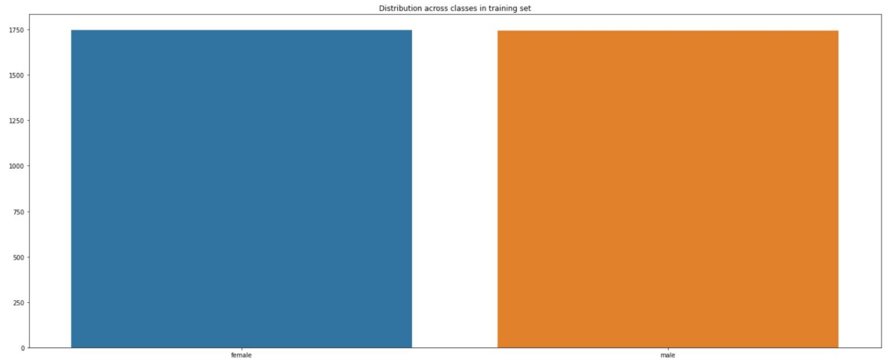


Figure 1. **Class and Image Count Graph**

The owner also developed an image cropping function using MTCNN to crop all the images only to keep the face. An image duplicate detector was also created so that it could eliminate any of the training images from appearing in the test or validation images.

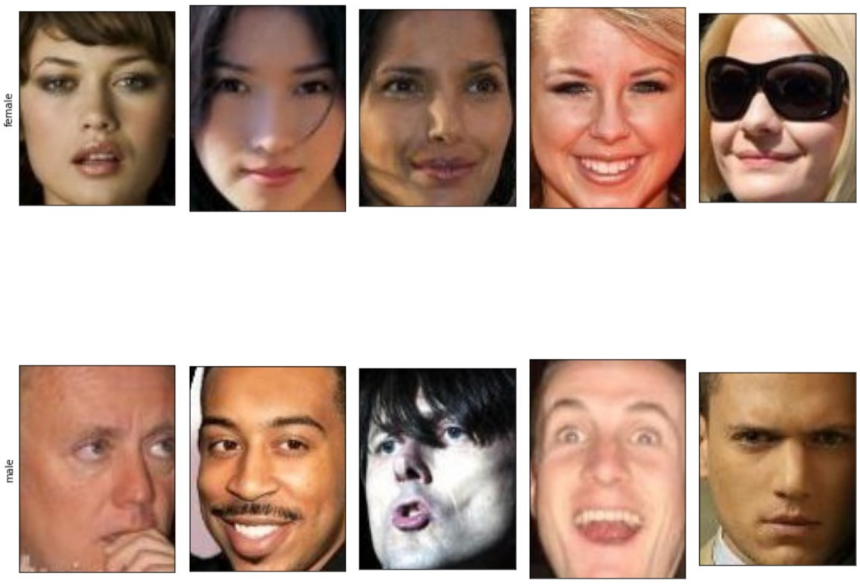


Figure 2. **Random Images from Dataset**

The MobileNet CNN model was used by the creator of the dataset because it was less computationally expensive and gave excellent result. As a beginner, two of the most common architecture ResNet and VGGNet was applied first on this dataset to get an idea how much accuracy it achieves. Data augmentation was also done to some extent to change the viewing perspective by zooming, whitening, shifting the width, height and many more. Finally, from the previous models output, a sequential model was created to test the accuracy whether it can beat the current one.

2. Related Works

On the paper by Janahiraman and Subramaniam et.al.[1] wanted to use multiple CNN architecture models to make gender categorization. Face pictures of Malaysians and some Caucasians were used to generate a dataset. The VGG-16 model had an accuracy of 88%, the ResNet-50 model had an accuracy of 85%, and Mobile Net model had an accuracy of 49%.

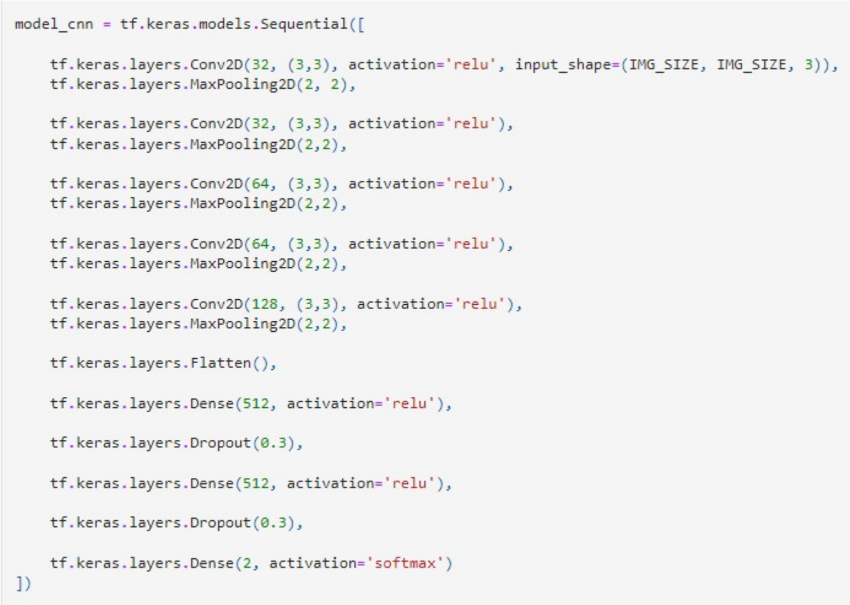
Using the Local Recipient AreasExcessive Learning Machine (LRA-ELM) and CNN architecture, Akbulut et.al.[2] accomplished gender recognition from facial photos. For age and gender recognition, the tests used around 11 thousand photos from the Adience Dataset (Eidinger et. al., 2014), with

LRA-ELM and CNN, the suggested technique achieved an accuracy of 80% and 87.13%, respectively. A simple convolutional network design was presented in (Levi and Hassner, 2015) et.al.[3] to improve the performance of automatic age and gender categorization. Even in the presence of insufficient training data, this method produces ideal outcomes. On the Adience database, this model produced high classification accuracy (Eidinger et. al., 2014)
Abdalrady and Aly proposed et.al.[4] the interchange of traditional CNN models with the PCANet model for gender categorization in (Abdalrady and Aly, 2020). In addition, PCANet decreased the size of the network design in sophisticated CNN models. For gender categorization, this approach has accuracy of 89.65

After analyzing these papers, a gender categorization system based on deep learning techniques is proposed in this paper. For gender categorization from photos, the custom CNN models will be used. The remainder of the paper is laid out as follows. The suggested approach is described in depth in the next section. After that, there is comparison of results achieved from different models and from custom CNN model.

3. Proposed Model

The model that has been implement for the project, is a custom CNN model. CNN is a strong neural network and one of the deep learning approaches. It's commonly utilized to solve difficulties in fields like Computer Vision and Image Processing. In each layer, CNN may substitute input data with trainable parameters and make correct guesses about the pictures' nature. Convolution layer, activation layer, pooling layer, fully connected layer, and dropout are the five primary types of neural layers in CNN architecture. Each layer type has a distinct function. The input volume is converted into an output volume of neuron activity by each layer of CNN, which is then sent to fully linked layers. While the early layers get basic characteristics such as edge information, the deep layers obtain more complex features that represent the image. The operations performed on the proposed custom CNN layers are described in detail in the following section.



```
model_cnn = tf.keras.models.Sequential([

    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(IMG_SIZE, IMG_SIZE, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),

    tf.keras.layers.Dense(512, activation='relu'),

    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(512, activation='relu'),

    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(2, activation='softmax')
])
```

Figure 3. Sequential Model details

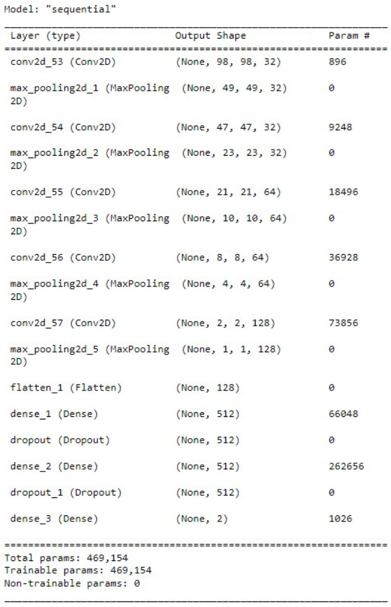
Convolutional Layers: CNN is built on top of this layer. The change is accomplished by rotating a filter over the picture that can be of various sizes, such as 3*3, 2*2. For this case, the used filter size was 3*3, because it is one of the most used size for image processing. It will always provide a third dimensional image while using 2D convolutions on pictures. The number of channels in the input picture determines this filter. As images of the dataset were colored images, so the number of channels were 3. That's why the suggested filter size was declared 3*3. For making the model efficient, total of 5 convolutional layers was used so that the training time can be reduced.

Activation Function: Because of the mathematical processes done at the convolutional layer, the network has a linear structure. The network acquires a non-linear structure as a result of the activation functions used in the activation layer. As a result, the network may learn quicker. In a neural network design, selecting activation function is critical. In this scenario, as the dataset is dealing with multiclass classification problem, so the perfect activation function for the output dense layer was "Softmax". Also, for the hidden layers, "ReLU" was used.

Pooling Layers: The pooling layer is the layer in CNN systems that is utilized to reduce size. The size reduction procedure may result in information loss, yet these losses are helpful to the network. Because the smaller size reduces the computational demand on the network's convolutional layers, it helps to prevent network overfitting. In this case, Max Pooling is used to aid over-fitting by giving an abstracted representation of the data. It also lowers the computational cost by minimizing the number of parameters that must be learned and gives basic translation to the internal representation.

Dense Layers: All activation in the preceding layers are fully linked to the neurons in this layer. Two-dimensional feature maps are converted into one-dimensional feature vectors as a consequence of these layers. The generated output can be used to classify a set of objects into a set of categories or as a feature outcome for further processing. In this proposed model, there are 2 fully connected dense layers were declared with 512 neurons and the final dense layer decides the output with the help of Softmax activation function to categorize the dataset into 2 different categories or classes.

Dropout: In deep learning, it is one of the most often utilized networking strategies. When CNN is trained with huge data, the network may become overfit. The machine starts to memorize the pattern for each iteration. With dropout, memorization is prevented by the underlying logic of eliminating some nodes from the network. Here, while training the model, for each dense layer, 30% neurons were dropped with the help of dropout so that the machine can learn more efficiently.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_53 (Conv2D)          (None, 98, 98, 32)        896

 max_pooling2d_1 (MaxPooling  (None, 49, 49, 32)       0
 2D)

 conv2d_54 (Conv2D)          (None, 47, 47, 32)        9248

 max_pooling2d_2 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 conv2d_55 (Conv2D)          (None, 21, 21, 64)        18496

 max_pooling2d_3 (MaxPooling  (None, 10, 10, 64)       0
 2D)

 conv2d_56 (Conv2D)          (None, 8, 8, 64)          36928

 max_pooling2d_4 (MaxPooling  (None, 4, 4, 64)         0
 2D)

 conv2d_57 (Conv2D)          (None, 2, 2, 128)         73856

 max_pooling2d_5 (MaxPooling  (None, 1, 1, 128)        0
 2D)

 flatten_1 (Flatten)         (None, 128)               0

 dense_1 (Dense)             (None, 512)               66048

 dropout (Dropout)           (None, 512)               0

 dense_2 (Dense)             (None, 512)               262656

 dropout_1 (Dropout)         (None, 512)               0

 dense_3 (Dense)             (None, 2)                 1026

=================================================================
Total params: 469,154
Trainable params: 469,154
Non-trainable params: 0
_________________________________________________________________
```

Figure 4. Model Summery and Layer details

4. Result and Discussion

For comparison of our model's outcome with other popular CNN models, ResNet50 and VGGNet16 was selected. As the ResNet50 has the most hidden layers, it produced the highest accuracy of 97.52%. On the other hand, VGG16 resulted an accuracy of 88.75%. The newly proposed model resulted 95.44% training accuracy, crossing the VGGNet16's training accuracy. The ResNet50 gave the highest accuracy for the image augmentation that were done. As the model had a less number of convolutional layers, the training ran smoothly and faster. As there were validation dataset also available, a test was also done with the model and got the validation accuracy of 95.31% for ResNet50, 86% for VGGNet16. For the proposed model the validation accuracy was about 92.50% which also crossed the VGGNet16's validation accuracy.

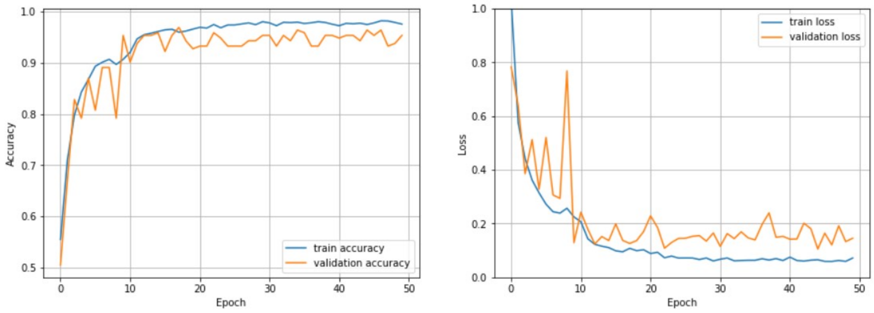


Figure 5. ResNet50 Result Graph

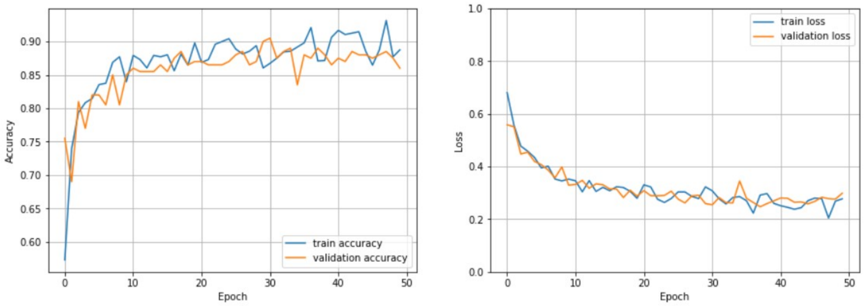


Figure 6. **VGGNet16 Result Graph**

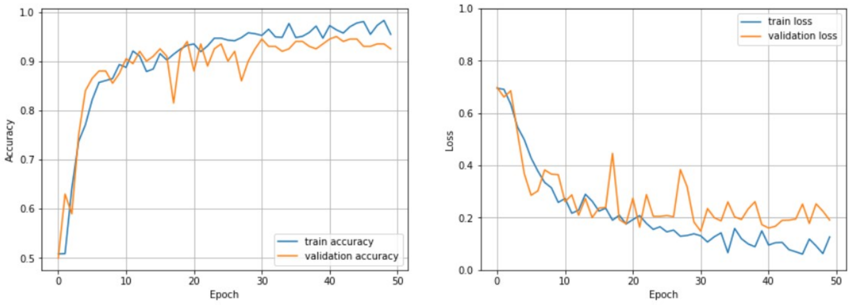


Figure 7. **Custom CNN Result Graph**

Table 1. **Models Results**

Model	1st Epoch Accuracy	25th Epoch Accuracy	50th Epoch Accuracy
ResNet50	55.52%	97.37%	97.52%
VGG16	57.29%	90.42%	88.75%
Custom CNN	50.86%	94.66%	95.44%

The total number of parameters for ResNet50 was 23,591,810 and for the VGGNet16, it was about 14,718,786. But for the proposed model, the number of total parameters was reduced to 469,154 only. The model was faster because it did not use the raw images of the data. Instead, it uses the augmented data. The graph also indicates the model was quite successful as the validation accuracy goes up with the training accuracy and the validation loss came down to almost less than 20%.
In this experiment we have also visualize a filter how a filter is learning and process an image for classification. The filter visualization is run in VGG16 layer. Here is some filter visualization images that show how a model is learning itself.

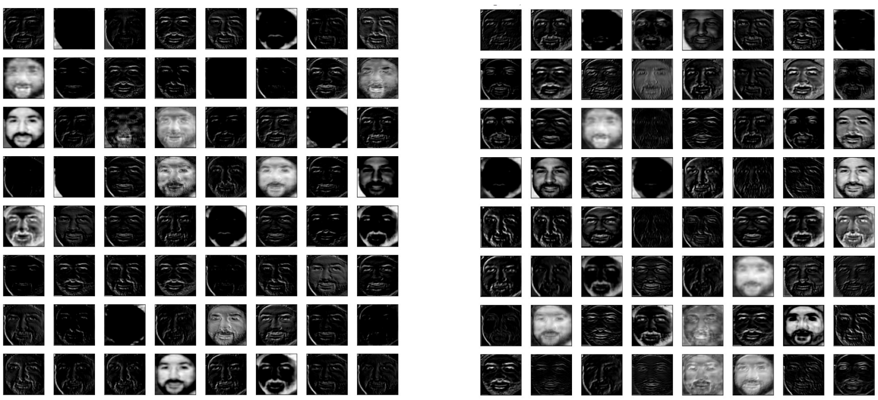


Figure 8. VGG16 Filter Visualization in a Layer

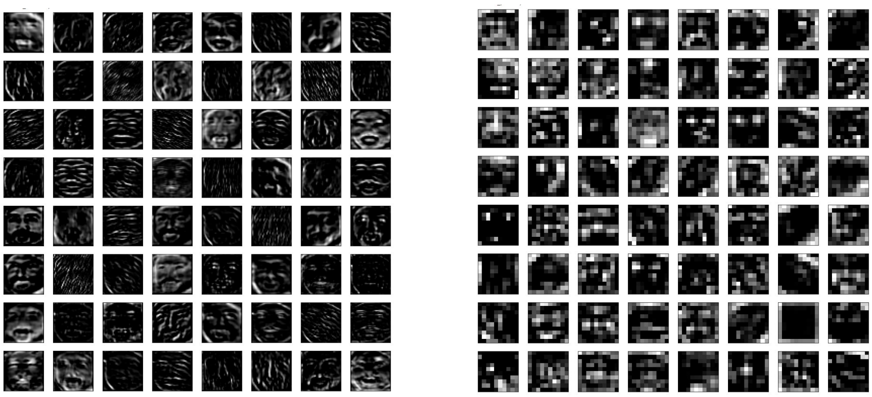


Figure 9. VGG16 Filter Visualization in a Layer

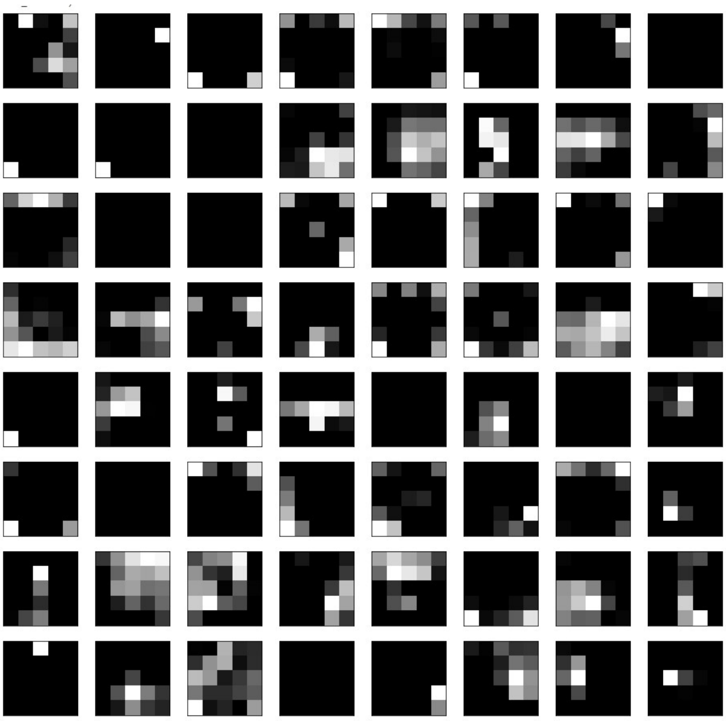


Figure 10. VGG16 Filter Visualization in a Layer

After VGGNet16, filter visualization was also done on the proposed model.

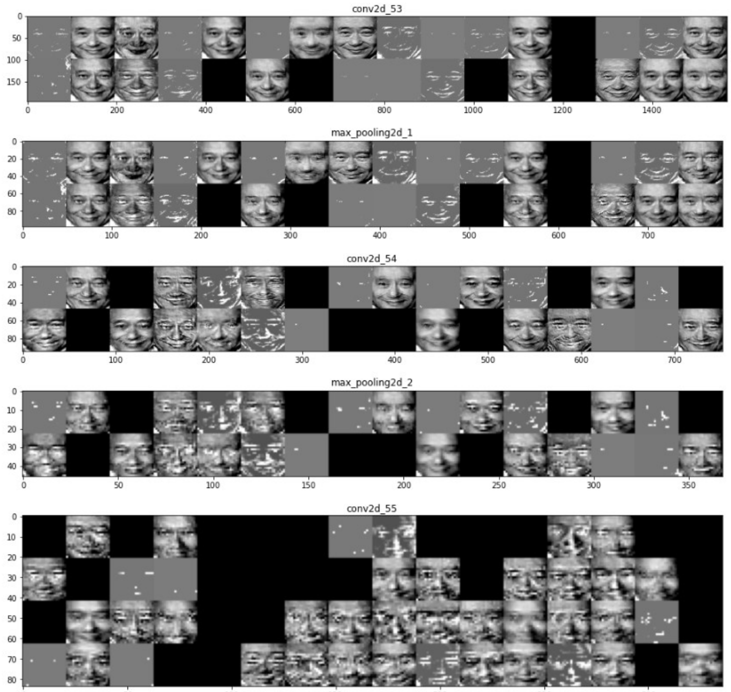


Figure 11. **Custom CNN Filter Visualization**

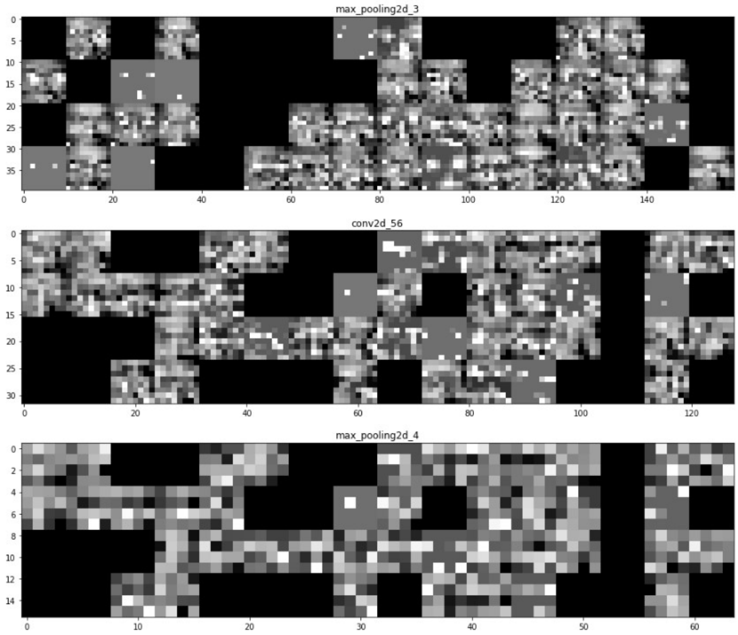


Figure 12. **Custom CNN Filter Visualization**

5. Conclusions

For gender categorization, a custom CNN deep learning model is proposed in this work. The model was mostly inspired from the AlexNet architecture. Some modifications has been done to train the model faster by reducing the number of parameters. The model is not so deep, but it provides a good level of accuracy than some predefined models like VGGNet16 and ResNet50. It can achieve higher accuracy if the image size can be increased. Due to the lack of proper GPU, the image size was reduced so that the model could run fluently. Further improvement can be done if more layers are added with proper image matrix calculation.

6. Dataset Information

Name: Gender Classification from an image
Link: https://www.kaggle.com/gpiosenka/gender-classification-from-an-image
Data: gender rev2
Size: 198MB
File: 11.8K files.

References

[**1**] Janahiraman, T. V., and Subramaniam, P., 2019, Gender Classification Based on Asian Faces using Deep Learning, In 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), 84-89.

[**2**] Akbulut, Y., Şengür, A., and Ekici, S., 2017, Gender recognition from face images with deep learning, In 2017 International artificial intelligence and data processing symposium (IDAP), IEEE, 1-4.

[**3**] Levi, G., and Hassner, T., 2015, Age and gender classification using convolutional neural networks, In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 34-42.

[**4**] Abdalrady, N. A., and Aly, S., 2020, February, Fusion of Multiple Simple Convolutional Neural Networks for Gender Classification, In 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), IEEE, 251-256.