Building Damage Detection based on Post-Hurricane Satellite Imagery using Transfer Learning and Convolutional Neural Networks

Presented To: Dr. Anu Sahni

Presented By: Aman Khanna (x19231938)
Jaswinder Singh (x19219997)
Khushboo Lavania(x19209835)

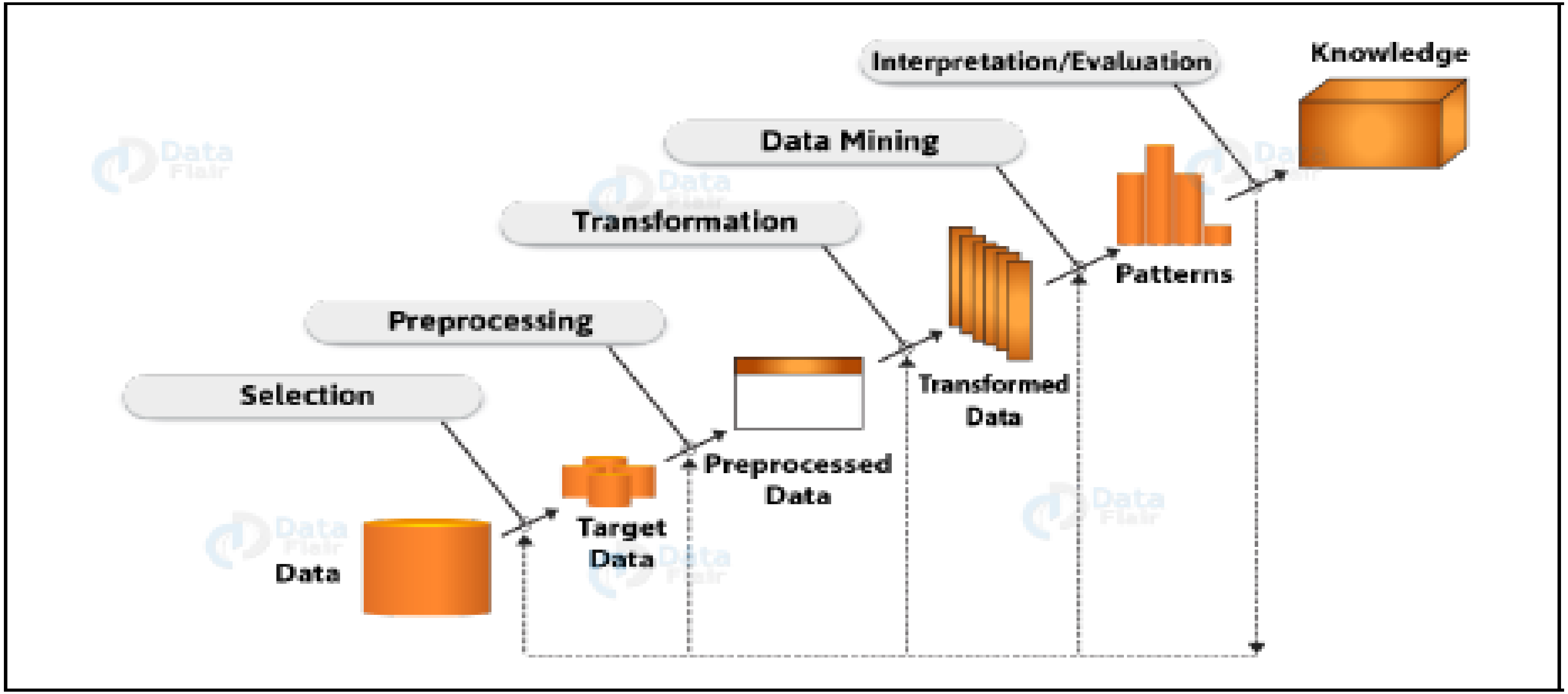
Objective

The objective of this study is to identify buildings that have been impacted by a natural disaster using satellite images of the areas impacted by the disaster.

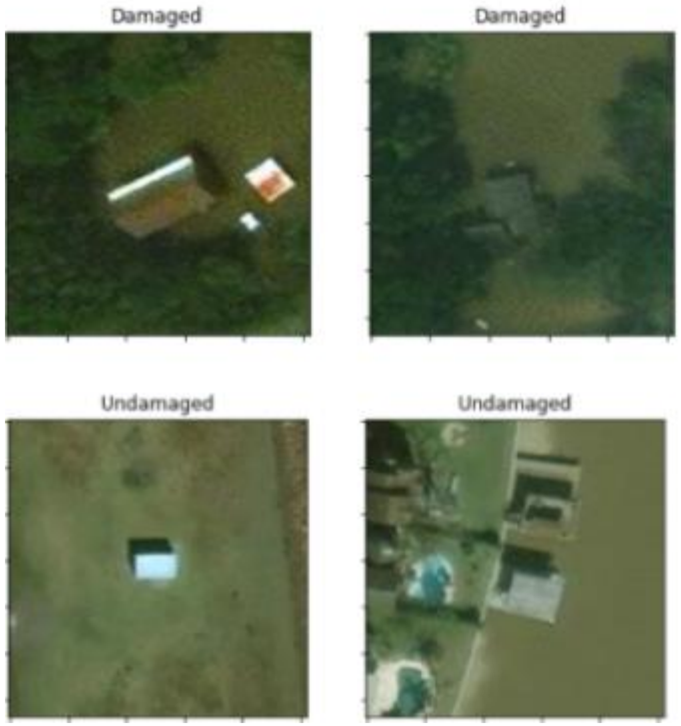
Related Works

After critically reviewing the key literature in the domain, we can conclude that different pre trained deep learning architectures like VGG, Inception V3, Resnet have performed exceptionally well in the satellite imagery classification tasks. They were even able to outperform the state-of-the-art modelsin some cases. Therefore, for the purpose of our study, we will also be using a pre-trained VGG16 architecture for the damage annotation in the optical satellite imagery. We will also design two different custom architectures using data augmentation and Leaky ReLU activation function and compare them based on different metrics like test and validation accuracy, ROC curves, precision, etc. The next section will discuss the methodology implemented for the study.

METHODOLOGY



Data Selection



Dataset Source

IEEE data portal

https://ieee-dataport.org/open-access/detecting-damaged-buildings-post-hurricane-satellite-imagery-based-customized)

Ethical Concerns

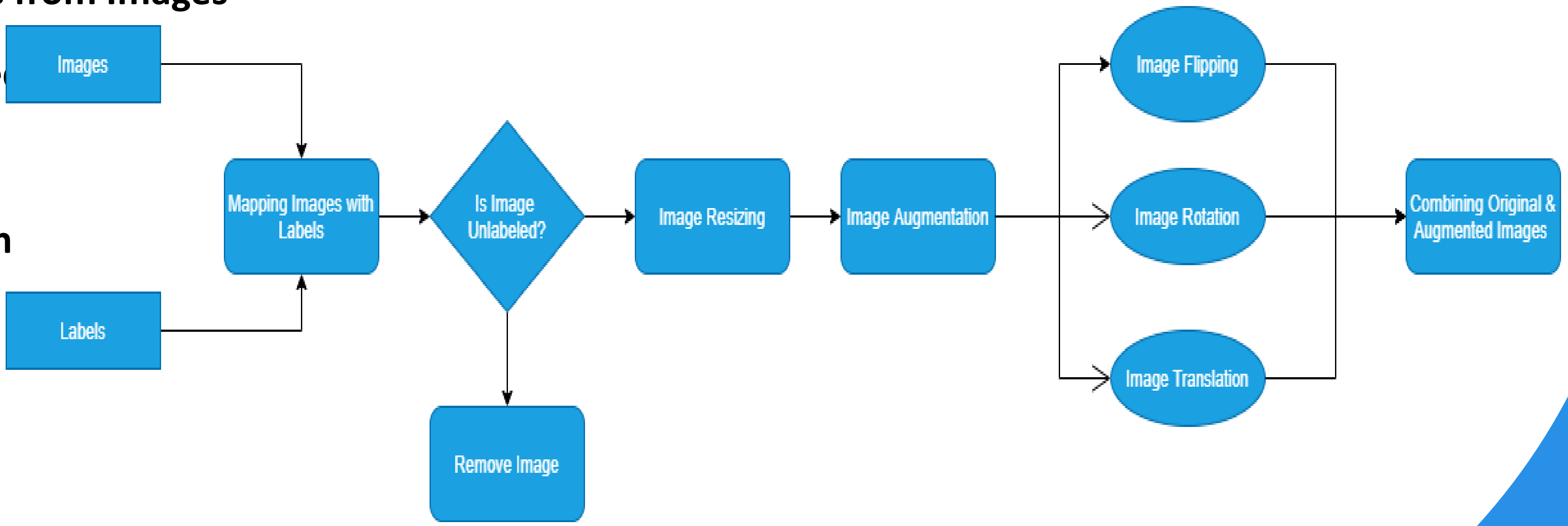
Data Pre-Processing & Transformation

- **Importing data into Google Colab**
- **Extracting Labels from Images**
- **Extracting Features from Images**
- **Handling Unlabelle**
- **Image Resizing**
- **Data Augmentation**

❑ **Image Flipping**

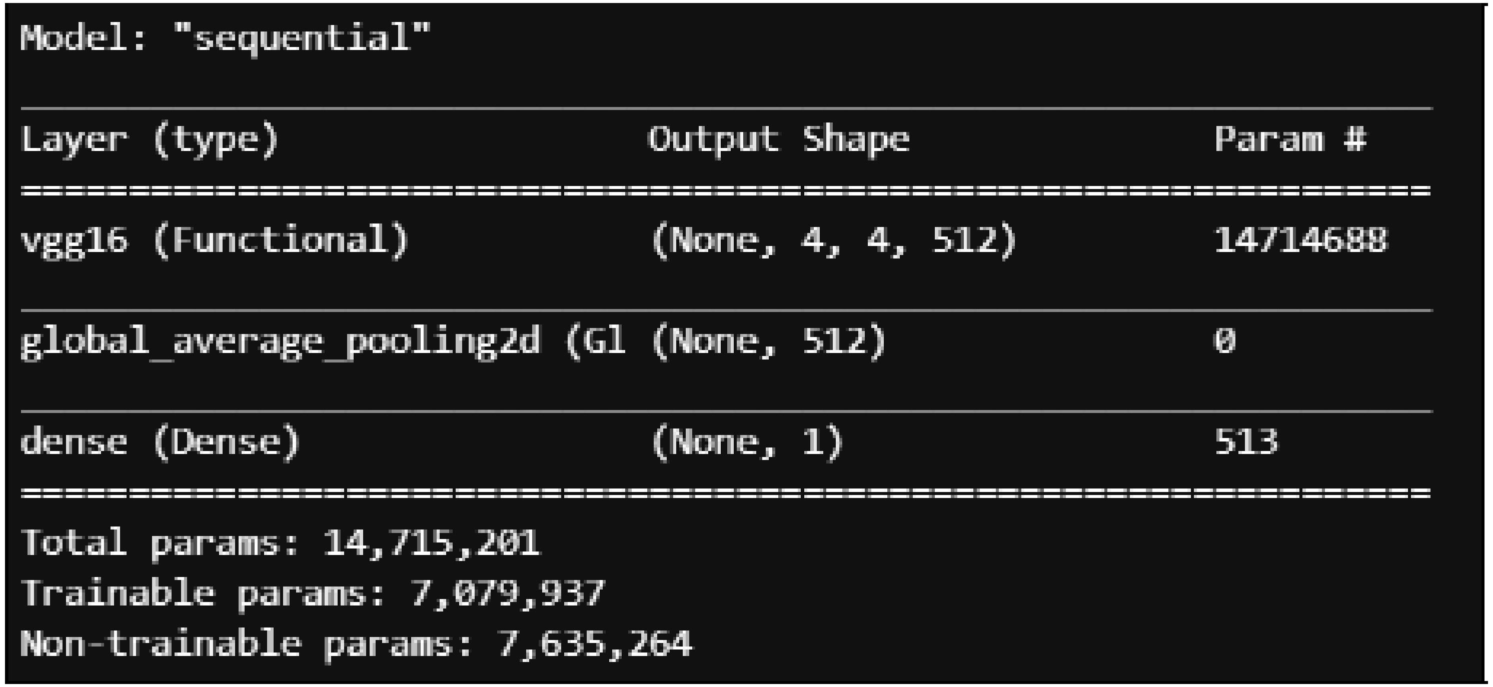
❑ Image Rotation

❑ **Data Translation**



Implementation and Evaluation

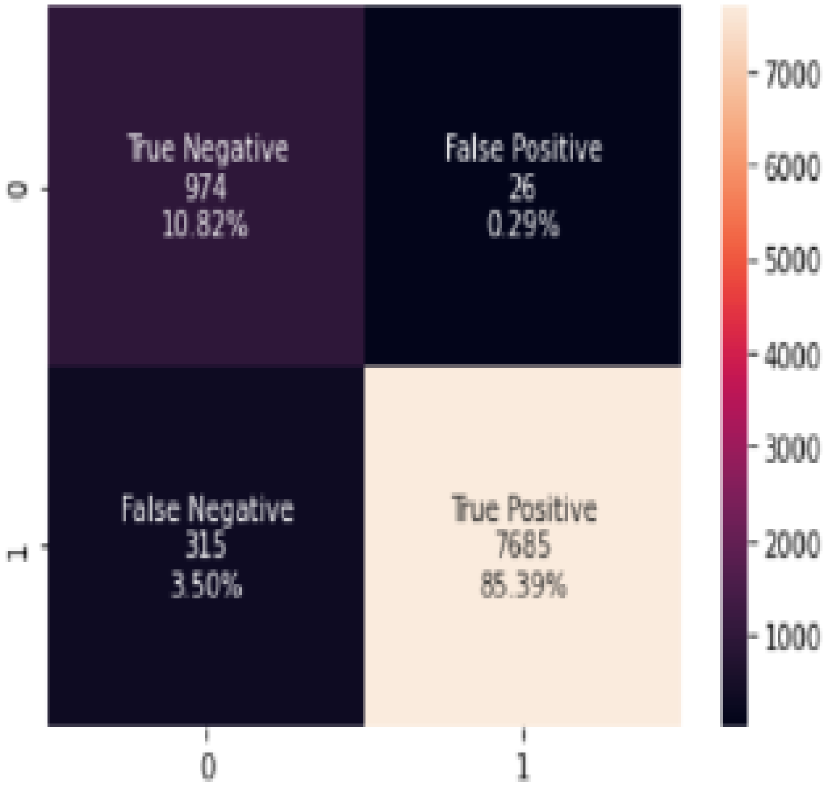
VGG16 (Transfer Learning)

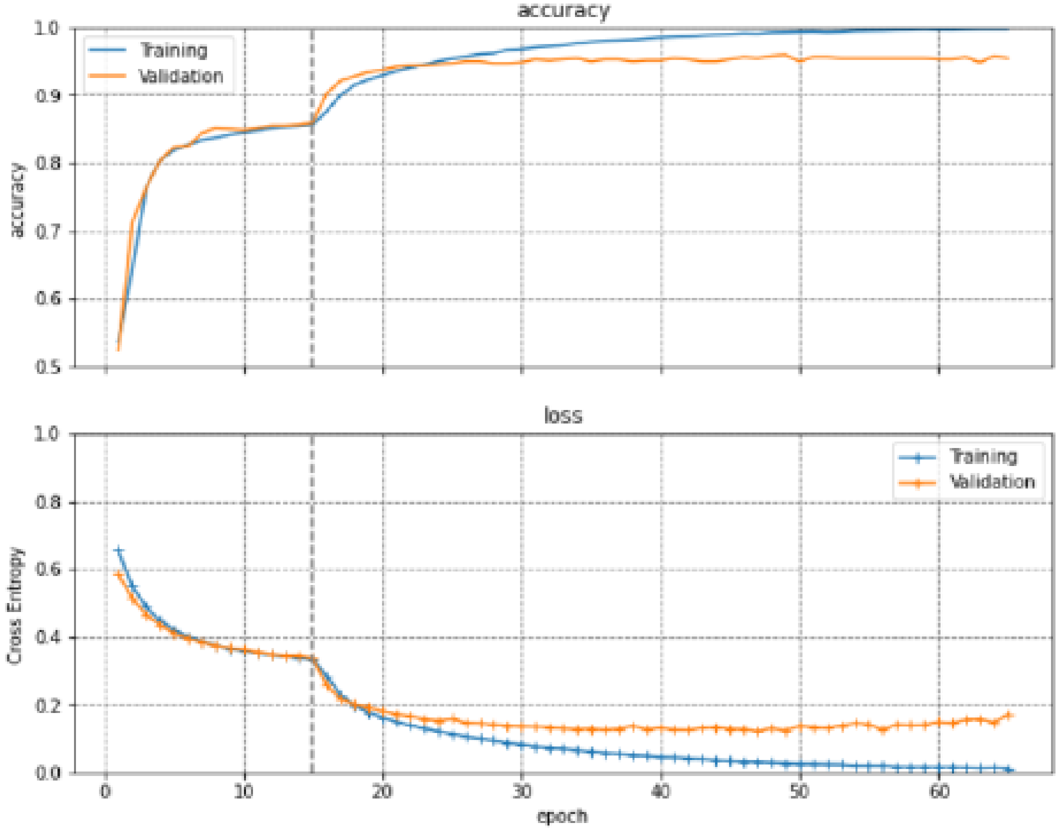


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 4, 4, 512)         14714688
_________________________________________________________________
global_average_pooling2d (Gl (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 513
=================================================================
Total params: 14,715,201
Trainable params: 7,079,937
Non-trainable params: 7,635,264
```

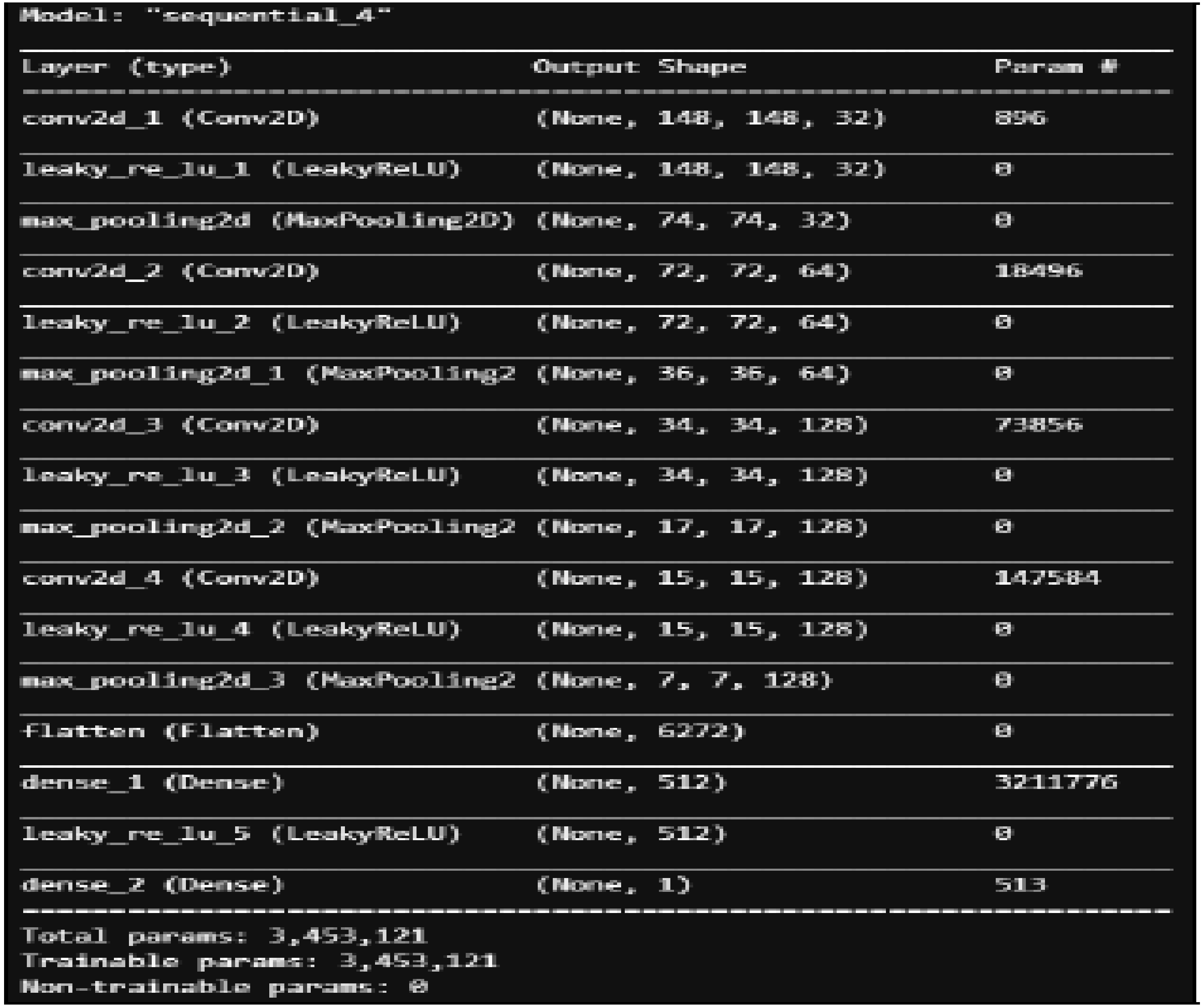
VGG16 Model Evaluation

- Accuracy - 96.21 %





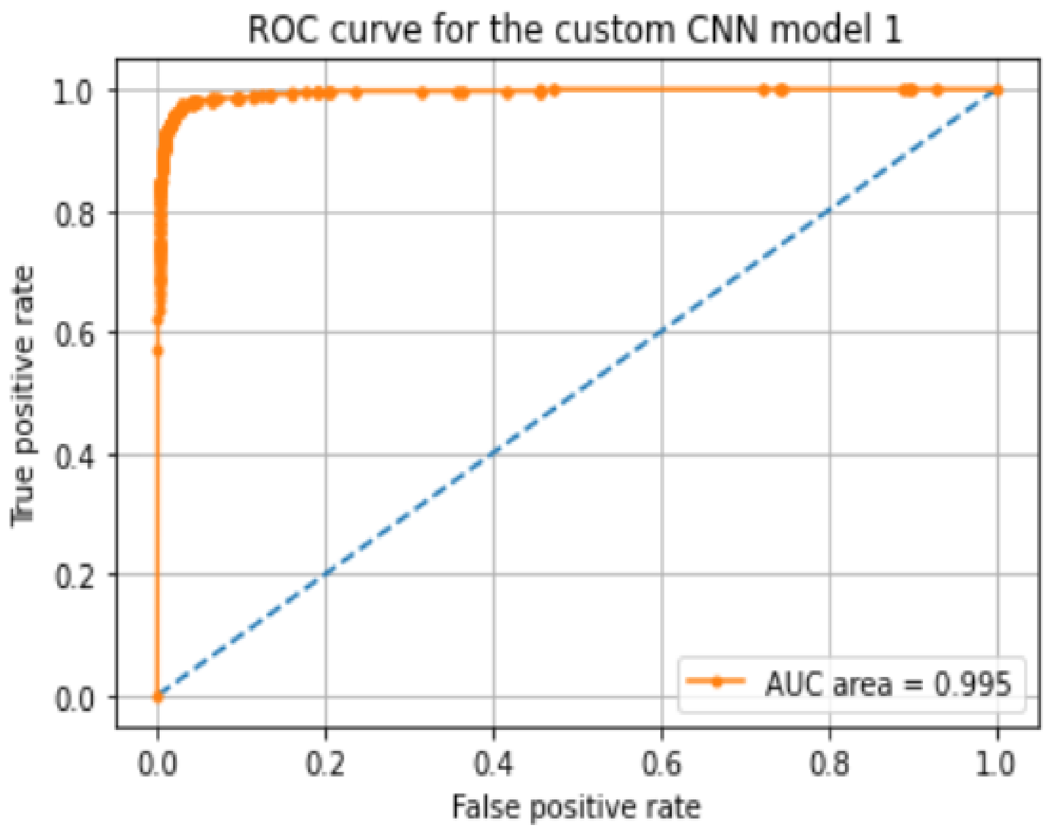
Custom Convolutional Neural Network 1

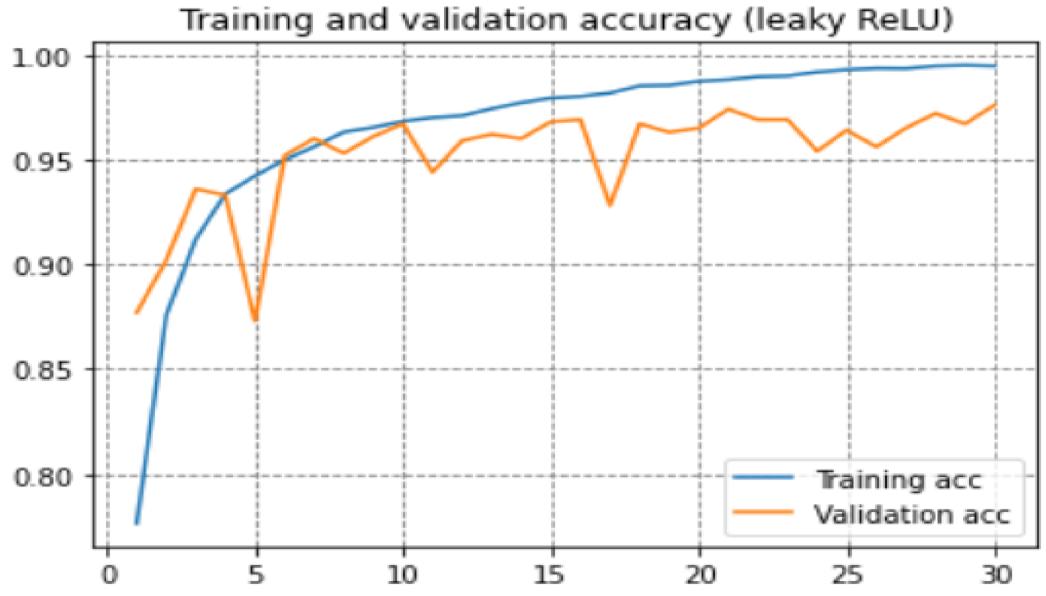


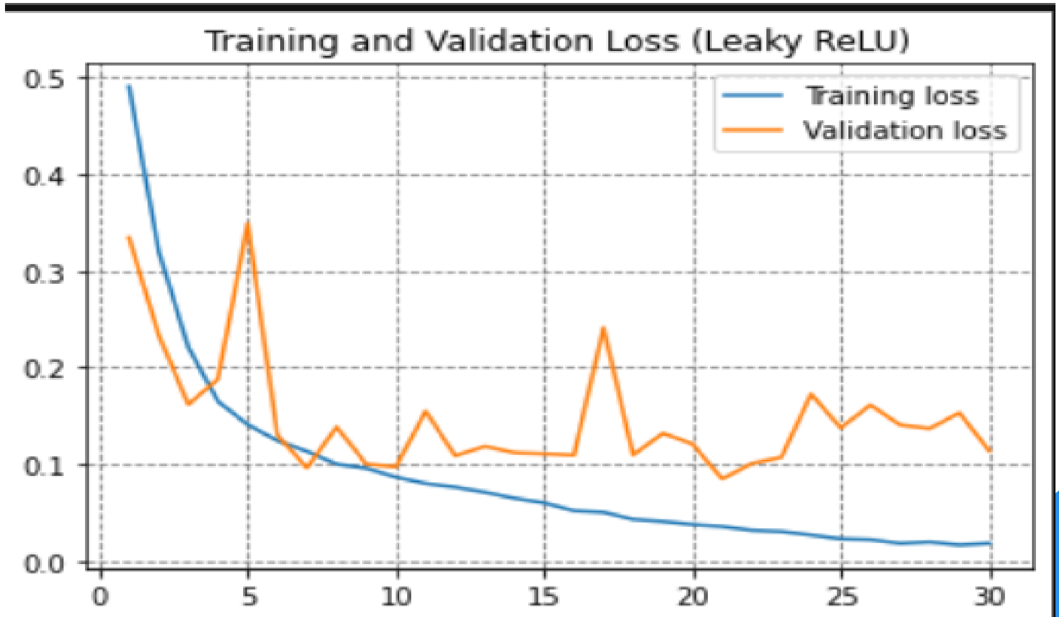
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 148, 148, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 72, 72, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 34, 34, 128)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 15, 15, 128)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3211776
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
```

Custom CNN 1 : Evaluation

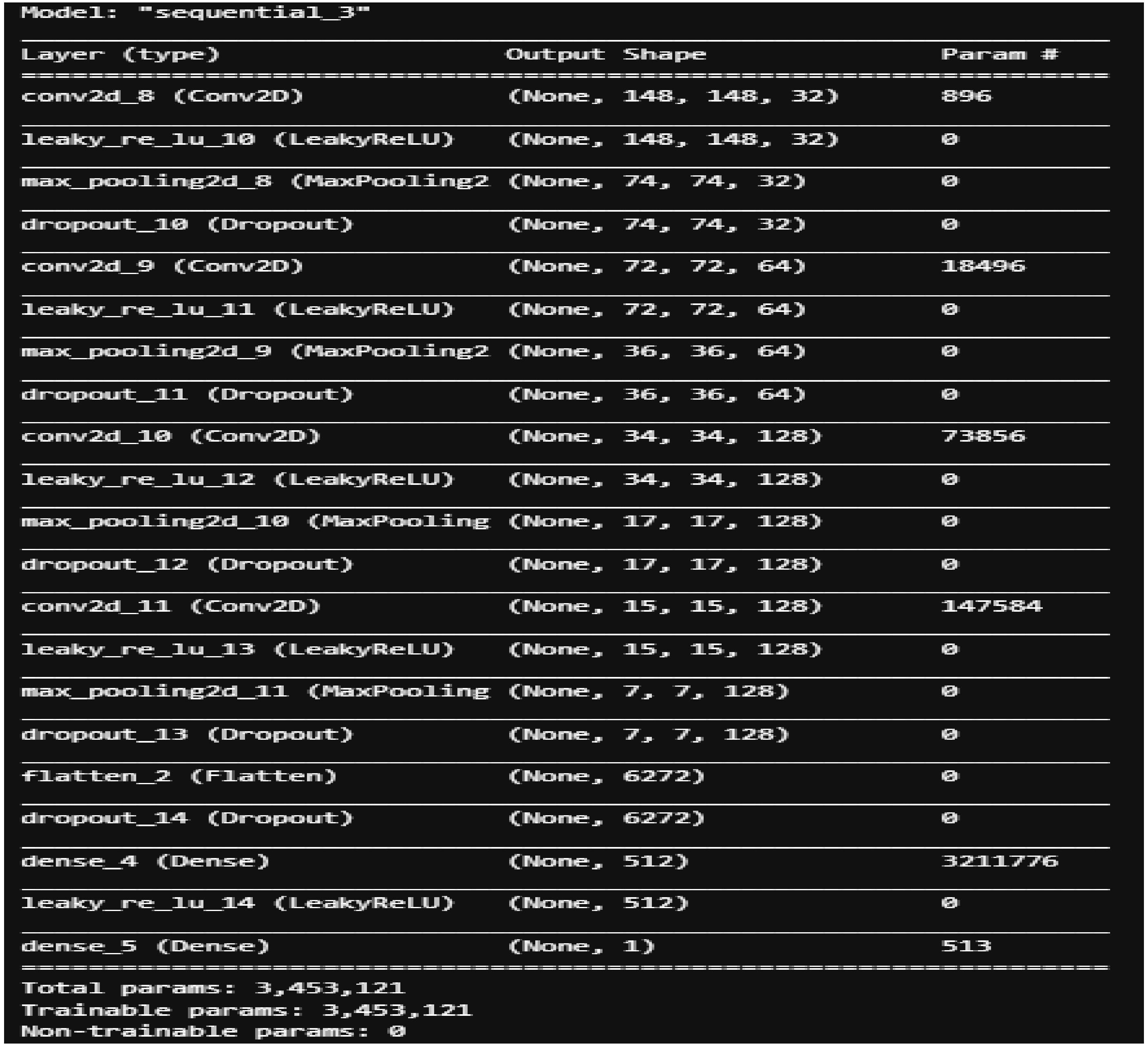
- **Accuracy : 89%**


ROC curve for the custom CNN model 1


Training and validation accuracy (leaky ReLU)


Training and Validation Loss (Leaky ReLU)

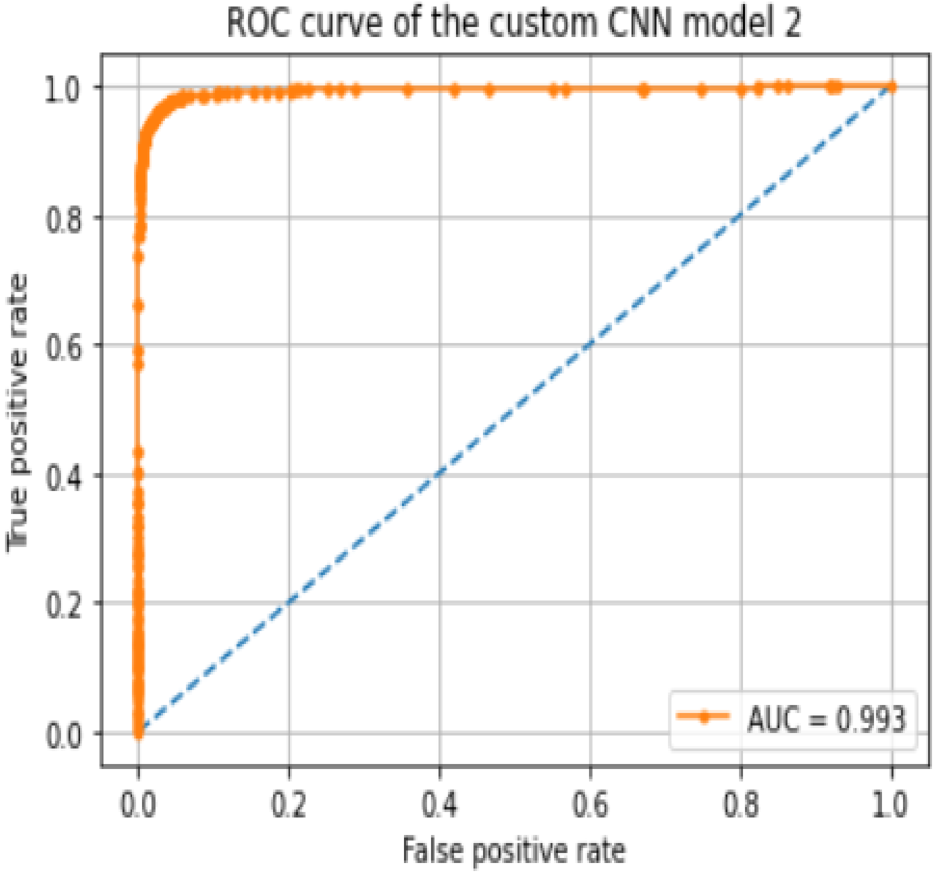
Custom Convolutional Neural Network 2

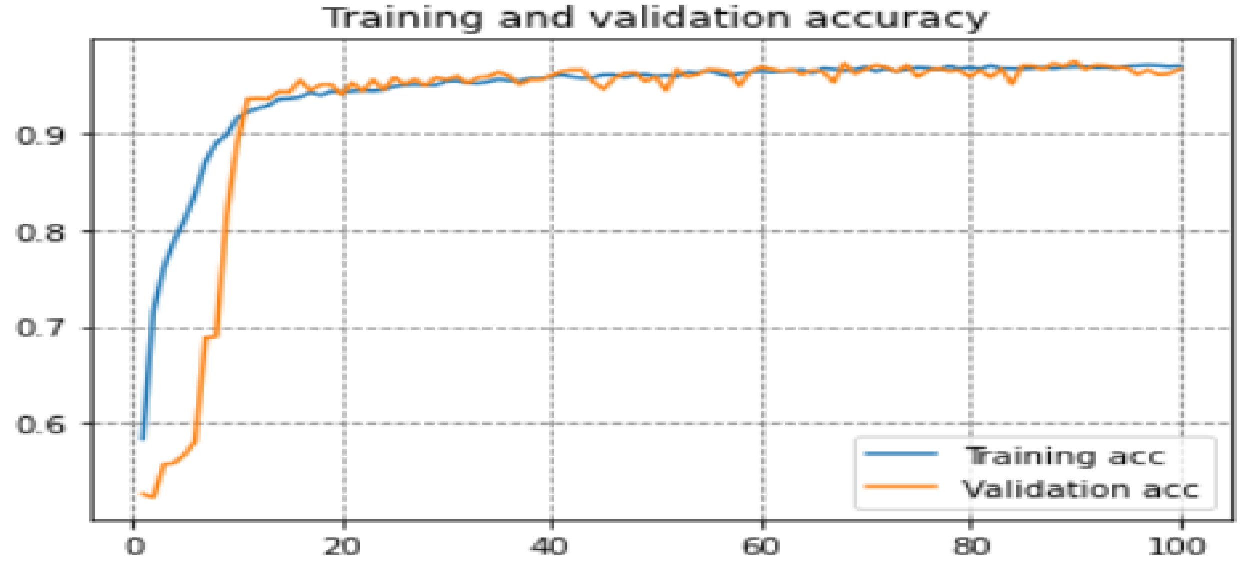


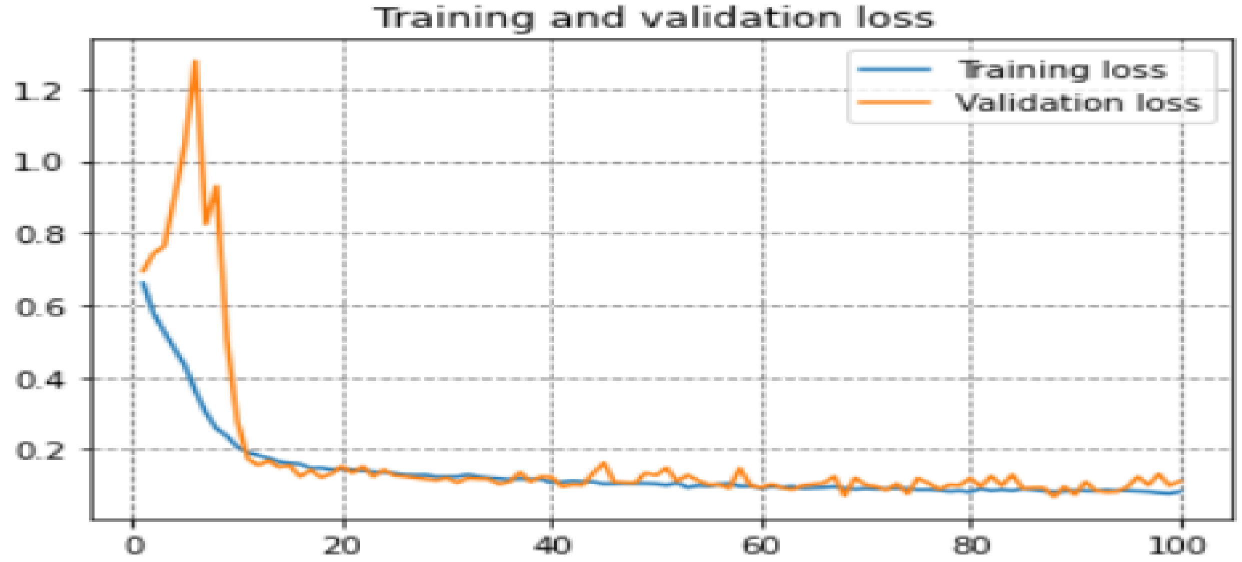
```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
leaky_re_lu_10 (LeakyReLU)   (None, 148, 148, 32)      0
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
dropout_10 (Dropout)         (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
leaky_re_lu_11 (LeakyReLU)   (None, 72, 72, 64)        0
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
dropout_11 (Dropout)         (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
leaky_re_lu_12 (LeakyReLU)   (None, 34, 34, 128)       0
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
dropout_12 (Dropout)         (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 15, 15, 128)       147584
_________________________________________________________________
leaky_re_lu_13 (LeakyReLU)   (None, 15, 15, 128)       0
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 7, 7, 128)         0
_________________________________________________________________
dropout_13 (Dropout)         (None, 7, 7, 128)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 6272)              0
_________________________________________________________________
dropout_14 (Dropout)         (None, 6272)              0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               3211776
_________________________________________________________________
leaky_re_lu_14 (LeakyReLU)   (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
```

Custom CNN 2 : Evaluation

- **Accuracy : 97.91**







Conclusion

The performance of Custom CNN 2 is better than VGG16 and Custom CNN 1 with an accuracy of 97.91.

ngiyabonga
danke
謝謝
teşekkür ederim
Благодаря
спасибо
dank je
thank you
gracias
tapadh leat
mochchakkeram
hvala
bedankt
dziękuję
go raibh maith agat
sukriya
kop khun krap
obrigado
arigato
takk
dakujem
grazie
terima kasih
мерси
감사합니다
merci