Abstract:

In this updated project, the Cifar100 dataset, which consists of one hundred categories, about five hundred images per category for the training set, was classified with a convolutional neural network. With data augmentation, the test accuracy increased from 0.43 to 0.59. Transfer Learning was also implemented, which resulted in astonishing high test accuracies. In the following report, the final model's architecture and other models that have been trained are presented.

Model's Architecture:

After once again, testing several models, each with different numbers of convolutional layers and different numbers of convolutional filters and parameters, the final model's architecture consists of six convolutional layers, each followed by a max-pooling layer, five dropout layers, and two dense layers followed by an output layer.

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_48 (Conv2D)           (None, 32, 32, 256)       7168

 batch_normalization_64 (Bat  (None, 32, 32, 256)       1024
 chNormalization)

 conv2d_49 (Conv2D)           (None, 32, 32, 256)       590080

 batch_normalization_65 (Bat  (None, 32, 32, 256)       1024
 chNormalization)

 max_pooling2d_24 (MaxPoolin  (None, 16, 16, 256)       0
 g2D)

 dropout_40 (Dropout)         (None, 16, 16, 256)       0

 conv2d_50 (Conv2D)           (None, 16, 16, 256)       590080

 batch_normalization_66 (Bat  (None, 16, 16, 256)       1024
 chNormalization)

 conv2d_51 (Conv2D)           (None, 16, 16, 256)       590080

 batch_normalization_67 (Bat  (None, 16, 16, 256)       1024
 chNormalization)

 max_pooling2d_25 (MaxPoolin  (None, 8, 8, 256)         0
 g2D)

 dropout_41 (Dropout)         (None, 8, 8, 256)         0

 conv2d_52 (Conv2D)           (None, 8, 8, 512)         1180160

 batch_normalization_68 (Bat  (None, 8, 8, 512)         2048
 chNormalization)

 conv2d_53 (Conv2D)           (None, 8, 8, 512)         2359808

 batch_normalization_69 (Bat  (None, 8, 8, 512)         2048
 chNormalization)
```

```
    max_pooling2d_26 (MaxPoolin   (None, 4, 4, 512)         0
                               g2D)

    dropout_42 (Dropout)          (None, 4, 4, 512)         0

    flatten_8 (Flatten)           (None, 8192)              0

  dense_24 (Dense)              (None, 1024)              8389632

    dropout_43 (Dropout)          (None, 1024)              0

   batch_normalization_70 (Bat   (None, 1024)              4096
                           chNormalization)

  dense_25 (Dense)              (None, 1024)              1049600

   batch_normalization_71 (Bat   (None, 1024)              4096
                           chNormalization)

    dropout_44 (Dropout)          (None, 1024)              0

   dense_26 (Dense)              (None, 100)               102500

=================================================================
                    Total params: 14,875,492
                Trainable params: 14,867,300
                Non-trainable params: 8,192
_________________________________________________________________
```

Figure 1 The Initial Model's Architecture

The trend of changes in train and validation loss and accuracy in the range of 80 epochs is as follows.

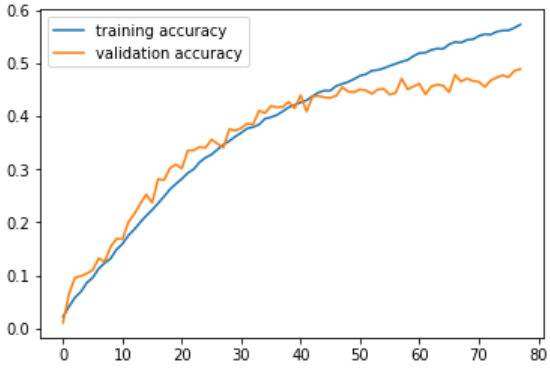


Figure 2 Training and Validation Accuracy of Final Model

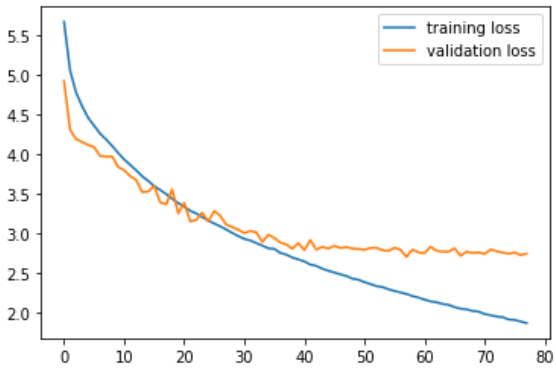


Figure 3 Training and Validation Loss of Final Model

And finally, with the use of data augmentation, the accuracy of the trained model on the test set was 59 percent.

```
Evaluate model on test data
78/78 [==============================] - 2s 25ms/step - loss: 1.5819 - accuracy: 0.5913
test loss, test acc: [1.5818543434143066, 0.5913461446762085]
```

Figure 4 Accuracy of Final Model on Test Set

Transfer Learning:

In this version of the project, two pre-trained networks were used to improve the test accuracy. ResNet50 and EfficientNetB0 were imported from keras.applications.

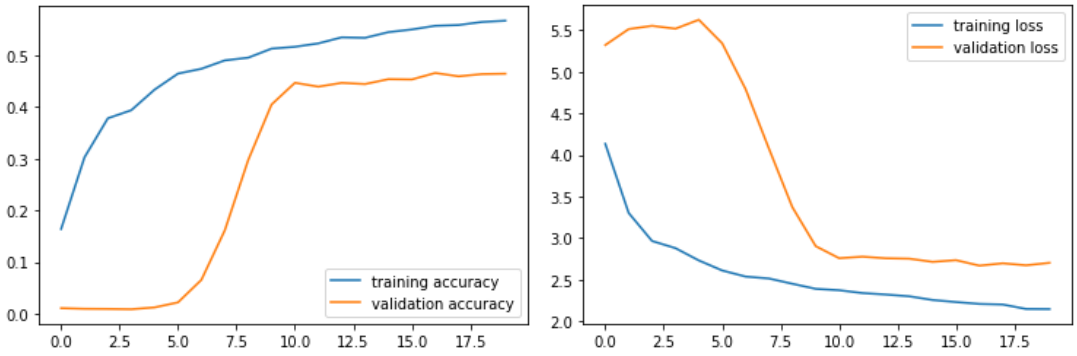
ResNet50:

The architecture of the network used is as follows:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 up_sampling2d (UpSampling2D  (None, 224, 224, 3)      0
 )

 resnet50 (Functional)       (None, None, None, 2048)  23587712

 global_average_pooling2d (G  (None, 2048)             0
```

```
                    lobalAveragePooling2D)

 dense (Dense)              (None, 512)               1049088

 batch_normalization (BatchN  (None, 512)             2048
                    ormalization)

 dropout (Dropout)          (None, 512)               0

 dense_1 (Dense)            (None, 100)               51300

=================================================================
                    Total params: 24,690,148
                Trainable params: 1,154,532
            Non-trainable params: 23,535,616
_________________________________________________________________
```

This model was trained for only 20 epochs due to the high number of parameters and, as a result, high training time. The trend of changes in train and validation loss and accuracy can be seen in the diagrams below.



And the test accuracy of this model is 67 percent, which can be improved if the number of epochs increases. However, because of the high number of parameters, we should be mindful of the risk of overfitting the model.

```
Evaluate model on test data
78/78 [==============================] - 8s 92ms/step - loss: 1.2966 - accuracy: 0.6787
test loss, test acc: [1.2965734004974365, 0.6786859035491943]
```

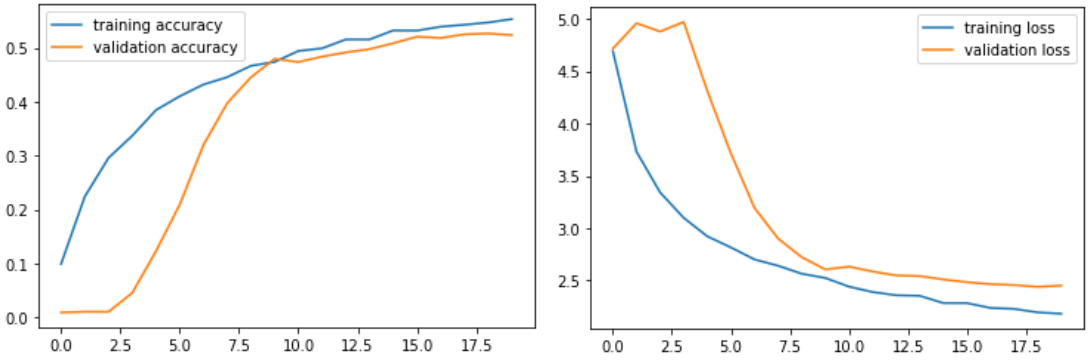
Figure 5 Test Accuracy on ResNet50

EfficeintNetB0:

The other pre-trained model used is EfficientNetB0, and the architecture is as follows.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 up_sampling2d_2 (UpSampling  (None, 224, 224, 3)      0
 2D)

 efficientnetb0 (Functional)  (None, None, None, 1280)  4049571

 global_average_pooling2d_2   (None, 1280)             0
 (GlobalAveragePooling2D)

 dense_4 (Dense)             (None, 512)               655872

 batch_normalization_2 (Batc  (None, 512)              2048
 hNormalization)

 dropout_2 (Dropout)         (None, 512)               0

 dense_5 (Dense)             (None, 100)               51300

=================================================================
Total params: 4,758,791
Trainable params: 750,212
Non-trainable params: 4,008,579
_________________________________________________________________
```

This model was also only trained on 20 epochs. The trend of changes in train and validation loss and accuracy can be seen in the diagrams below.



The test accuracy of this model is 68 percent, which can be improved again if trained for longer.

```
Evaluate model on test data
78/78 [==============================] - 6s 60ms/step - loss: 1.2337 - accuracy: 0.6847
test loss, test acc: [1.2337055206298828, 0.6846955418586731]
```

Figure 6 Test Accuracy on EfficientNetB0