Image Classification

Artificial Neural Networks and Deep Learning [SC]

Team: 59

Team Member	ID
مارينا نادي شحاتة عوض	20201700636
مايان محمد حلمي السيد	20201700646
منة الله محمد السيد احمد لاشين	20201701205
منه حاتم محمود حامد	20201701144
منة الله محمد مصطفي	20201700862
عائشة اشرف عز الدين	20201700428

Data Preparation

- The function reading_images() reads images from specified folders, resizing and normalize them, and preparing them for further processing.

- The create_label() creates a one-hot encoded vector from the numerical class label.

- Data Augmentation by ImageDataGenerator is set up to perform data augmentation (rotations, shifts, flips, etc.) for increasing dataset variety. The augmented images are concatenated with the original training set. The model is trained using augmented and original data through **model.fit**.

Model Training

VGG:

- Constructs a VGG-like model using **tflearn** by defining its layers with convolutional, pooling, and fully connected layers. (Alternating convolutional layers with max-pooling layers for downsampling and the fully connected layers at the end of the network perform classification)

Transformer:

- Patching: Divides input images into patches and applies Convolutional layers to extract features from these patches.
- Positional Encoding: Adds positional information to the image patches to preserve spatial relationships.
- Transformer Blocks: Utilizes multiple transformer blocks (Multi-Head Self-Attention, Feed-Forward Neural Networks, Classification Head)

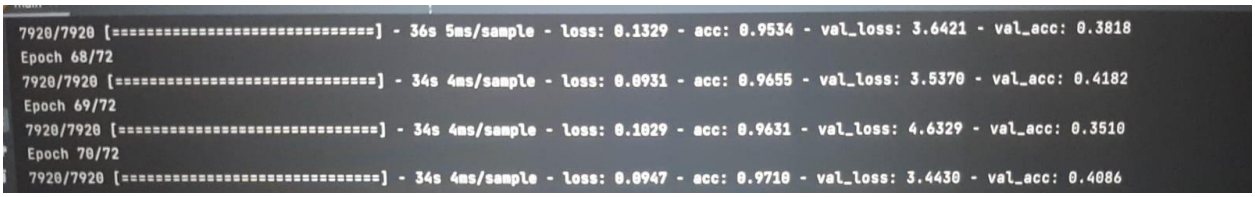
AlexNet Architecture using Keras:

- Model Definition: Defines an AlexNet architecture using Keras' Sequential API with Convolutional, MaxPooling, Flatten, Dense, and Dropout layers.
- Model Compilation and Training: Compiles the model with specified loss function (categorical_crossentropy), optimizer(SGD), momentum(0.9), and metrics(accuracy). Then trains the model using the training data.

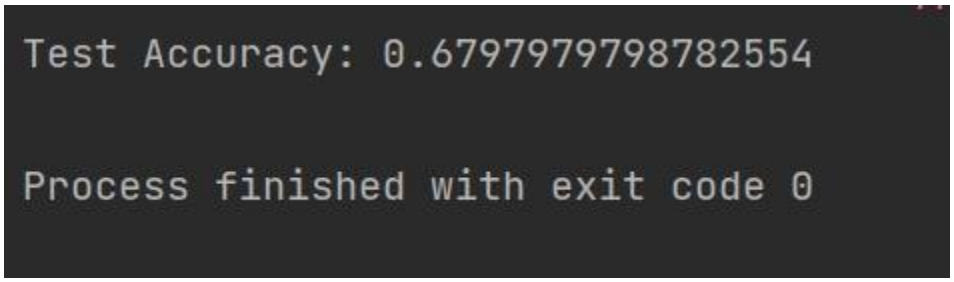
Evaluation

Evaluation

- **AlexNet:** after 70 epochs we got 40% accuracy.



- **VGG:**



- **Transformer:**

```
223/223 [==========] - 60s 270ms/step - loss: 1.5736 - accuracy: 0.2588 - val_loss:
1.6371 - val_accuracy: 0.2134
Epoch 41/50
223/223 [==========] - 60s 271ms/step - loss: 1.5733 - accuracy: 0.2570 - val_loss:
1.5776 - val_accuracy: 0.2664
Epoch 42/50
223/223 [==========] - 60s 270ms/step - loss: 1.5703 - accuracy: 0.2623 - val_loss:
1.7043 - val_accuracy: 0.1843
Epoch 43/50
223/223 [==========] - 60s 270ms/step - loss: 1.5707 - accuracy: 0.2639 - val_loss:
1.7263 - val_accuracy: 0.1843
Epoch 44/50
223/223 [==========] - 60s 270ms/step - loss: 1.5718 - accuracy: 0.2628 - val_loss:
1.6670 - val_accuracy: 0.1843
Epoch 45/50
223/223 [==========] - 60s 270ms/step - loss: 1.7268 - accuracy: 0.2715 - val_loss:
6.5597 - val_accuracy: 0.1843
Epoch 46/50
223/223 [==========] - 60s 270ms/step - loss: 1.8788 - accuracy: 0.2111 - val_loss:
1.6017 - val_accuracy: 0.2247
Epoch 47/50
223/223 [==========] - 60s 270ms/step - loss: 1.7918 - accuracy: 0.2291 - val_loss:
1.6111 - val_accuracy: 0.2134
Epoch 48/50
223/223 [==========] - 60s 270ms/step - loss: 1.7461 - accuracy: 0.2262 - val_loss:
1.6185 - val_accuracy: 0.2134
Epoch 49/50
223/223 [==========] - 60s 270ms/step - loss: 2.2427 - accuracy: 0.2141 - val_loss:
1.7834 - val_accuracy: 0.2260
Epoch 50/50
223/223 [==========] - 60s 271ms/step - loss: 2.2466 - accuracy: 0.2165 - val_loss:
3.0346 - val_accuracy: 0.2083
```

Conclusion

From the above evaluation w conclude that the VGG is the best model among these.