Age Classification

A CNN model

Nicholas Sanso

Problem Statement

- LinkedIn needs to verify the ages of individuals who use its website so that it can offer users advertisements to products that they will purchase and so that it can sell companies age demographic information.

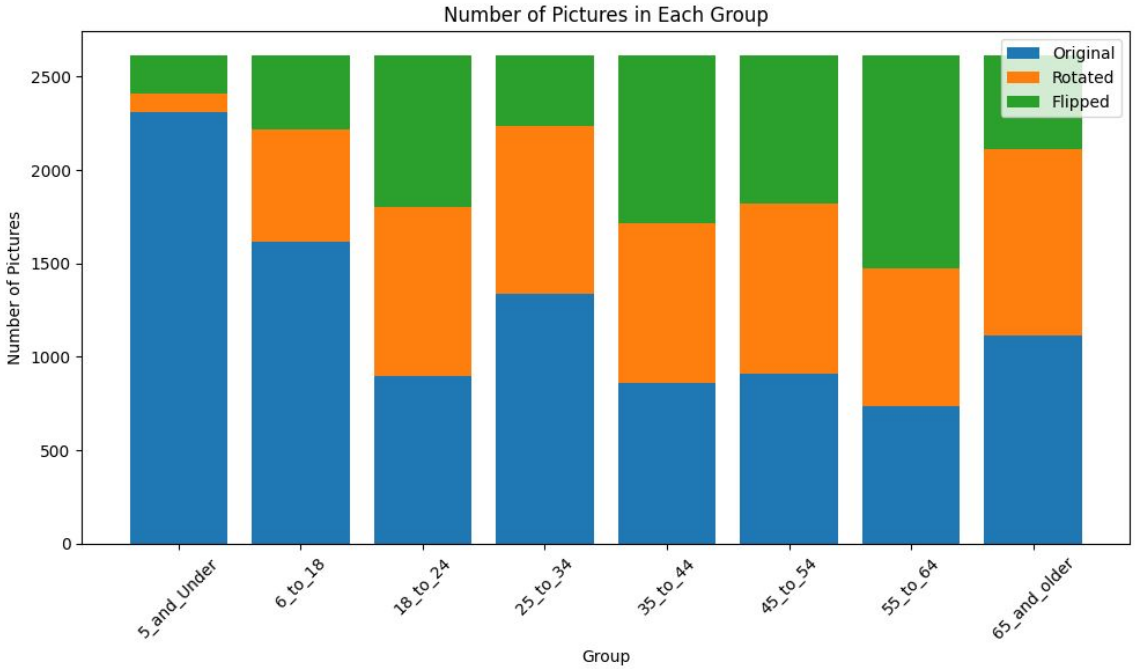
Model Selection

Convolution Neural Networks (CNN):

1. Convolutional layers apply filters to accentuate complex patterns
2. Pooling layers reduce the size of feature maps, decrease number of parameters and fitting time
3. CNN don’t connect to every previous neuron, reducing fitting time
4. Synergize with fully connected layers to learn global patterns from the convolutional layers

Preprocessing

- Implemented rotation and flipping techniques to balance classes
- Implemented techniques in all classes
- Downsample groups to balance subgroups

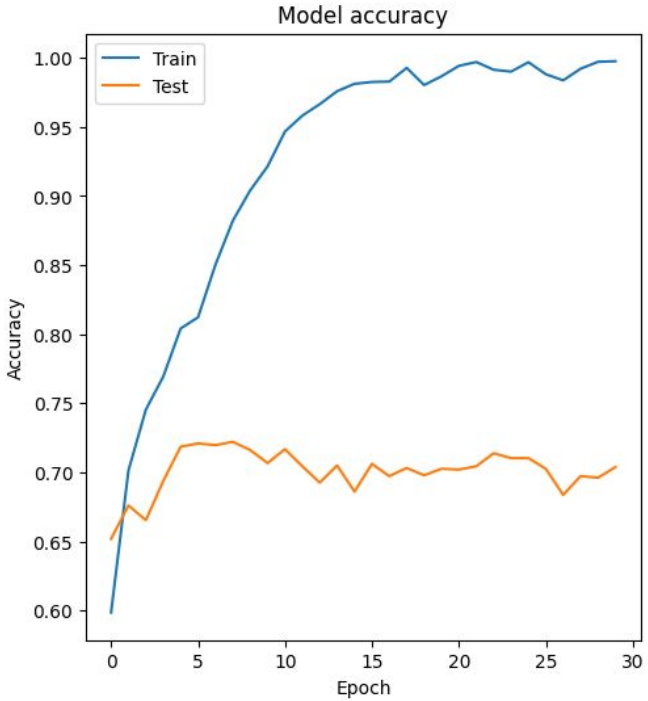


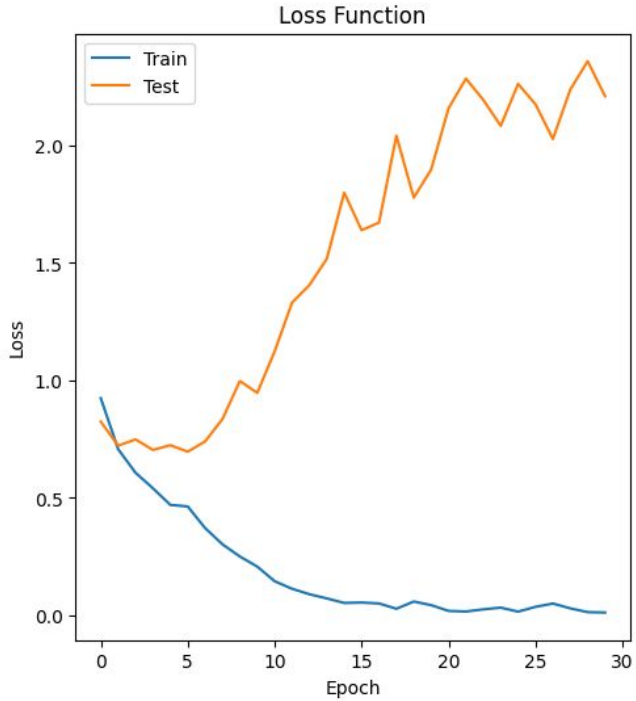
Model 1

```
model_1 = Sequential([
    Conv2D(8, (3, 3), activation='relu', input_shape=(200, 200, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(16, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(8, activation='softmax')
])
```

Model 1 Results

- Enormously overfit, batch size= 128
- Test accuracy stalls
- Solution: implement early stopping and regularization





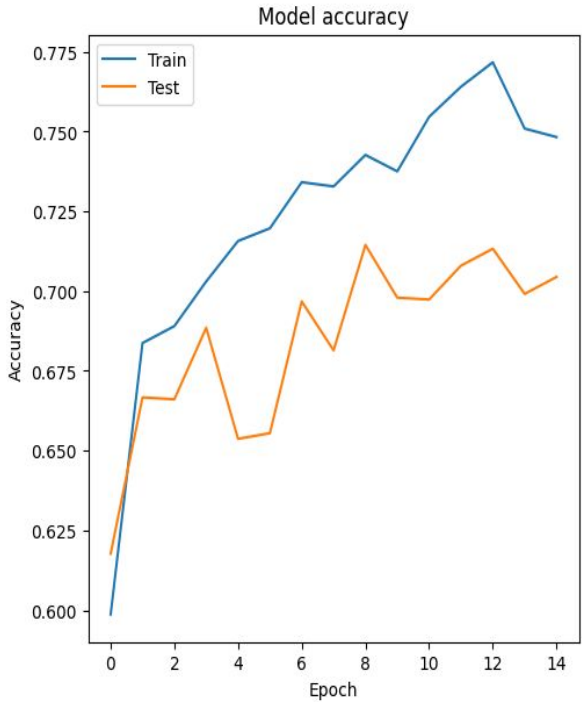
Model 2

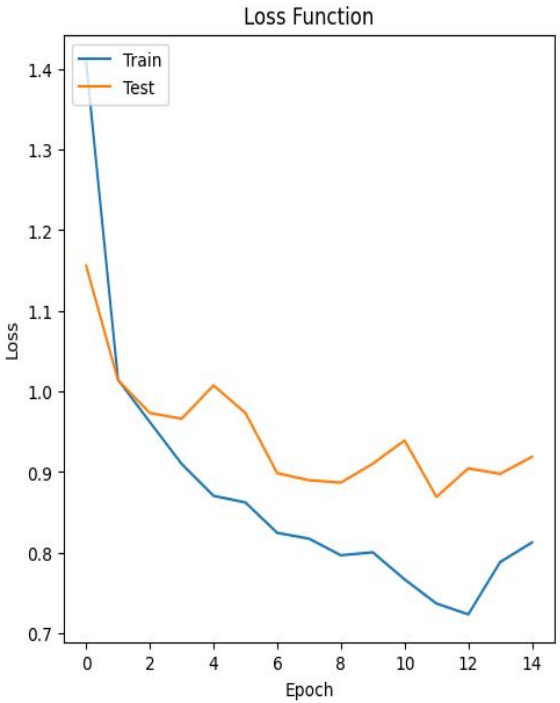
- Ridge Regularizer added.
- Batch size set to 32
- Early Stopping added

```
model_2 = Sequential([
    Conv2D(8, (3, 3), activation='relu', input_shape=(200, 200, 3), kernel_regularizer=l2(0.01)),
    MaxPooling2D((2, 2)),
    Conv2D(16, (3, 3), activation='relu', kernel_regularizer=l2(0.01)),
    MaxPooling2D((2, 2)),
    Conv2D(32, (3, 3), activation='relu', kernel_regularizer=l2(0.01)),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(64, activation='relu', kernel_regularizer=l2(0.01)),
    Dense(8, activation='softmax')
])

model_2.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss', patience=3)

model_2_history= model_2.fit(train_data, validation_data=val_data, epochs=30, callbacks=[early_stopping])
```

Model 2

- Substantial improvement
- Batch size set to 32
- Early Stopping a bit too tight





Model 3

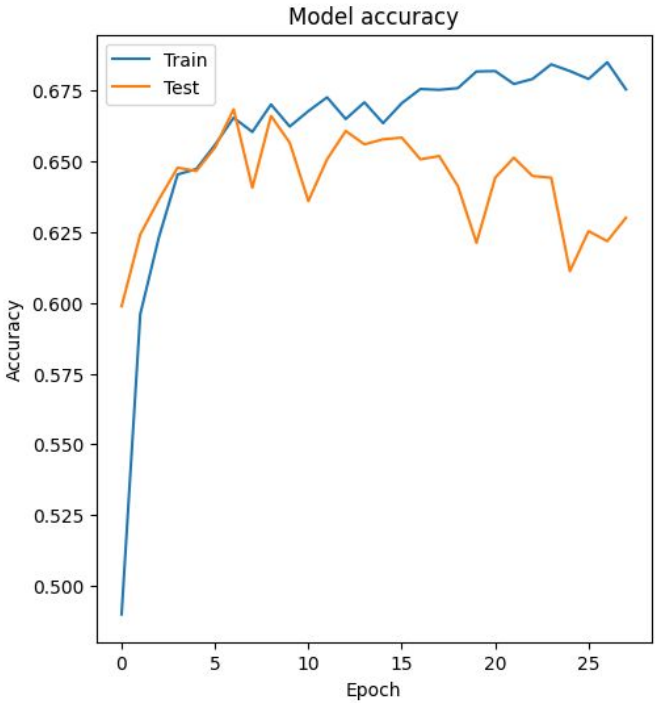
- Increase patience
- Add Dropout

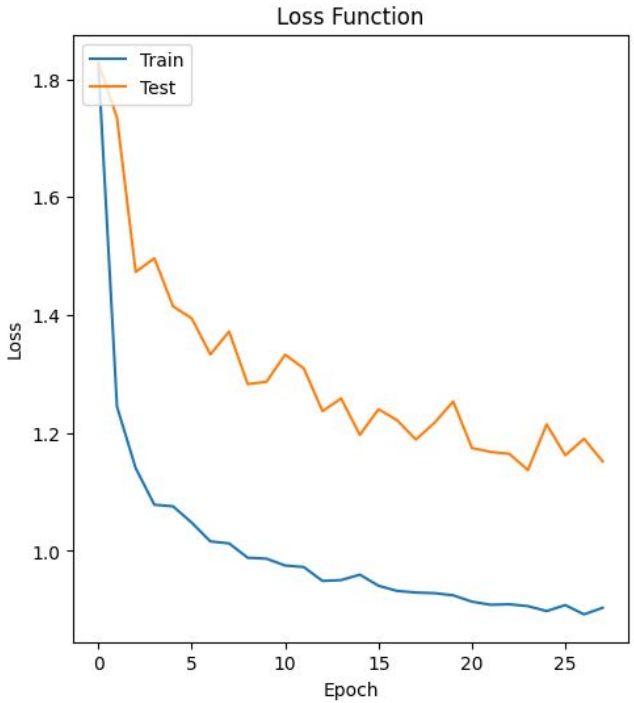
```
model_3 = Sequential([
    Conv2D(8, (3, 3), activation='relu', input_shape=(200, 200, 3), kernel_regularizer=l2(0.01)),
    Dropout(0.5),
    MaxPooling2D((2, 2)),
    Conv2D(16, (3, 3), activation='relu', kernel_regularizer=l2(0.01)),
    Dropout(0.5),
    MaxPooling2D((2, 2)),
    Conv2D(32, (3, 3), activation='relu', kernel_regularizer=l2(0.01)),
    Dropout(0.5),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(64, activation='relu', kernel_regularizer=l2(0.01)),
    Dropout(0
    Dense(8, activation='softmax')
])

model_3.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

early_stopping = EarlyStopping(monitor='val_loss', patience=4)

model_3_history = model_3.fit(train_data, validation_data=val_data, epochs=30, callbacks=[early_stopping])
```

activation: Any

Model 3

- Added Dropout layers
- Performance deteriorated from Model 2
- Learning dominated by a handful of neurons





Conclusions

- The increase in test loss function and drop in test accuracy with the inclusion of the Dropout function to the model architecture indicates that there are probably a limited number of patterns that the model relies heavily on.

- The improvement in performance with the reduction in batch size indicates that the model may have been converging to a local minimum when minimizing the loss function during the (training) gradient descent process.

- Considerations for model performance must be balanced against hardware requirements. MaxPooling2D’s contribution to reducing training time is indispensable for all CNN models.

Future Steps

- Transfer Learning to see if the age classifier could classify race, gender or other attributes commonly targeted by advertisers or pollsters
- Increase the total number of images in order to achieve a more proportional balance of the number of modified images
- Pull the weights of each of the neural networks to explore the standard deviation of the weights which will indicate the variety of features that the model is using to learn.
- 4. Rum more models to isolate other parameters in the model