Image Classification Using Street View House Numbers (SVHN) Dataset

Matthew Lim

June 8, 2022

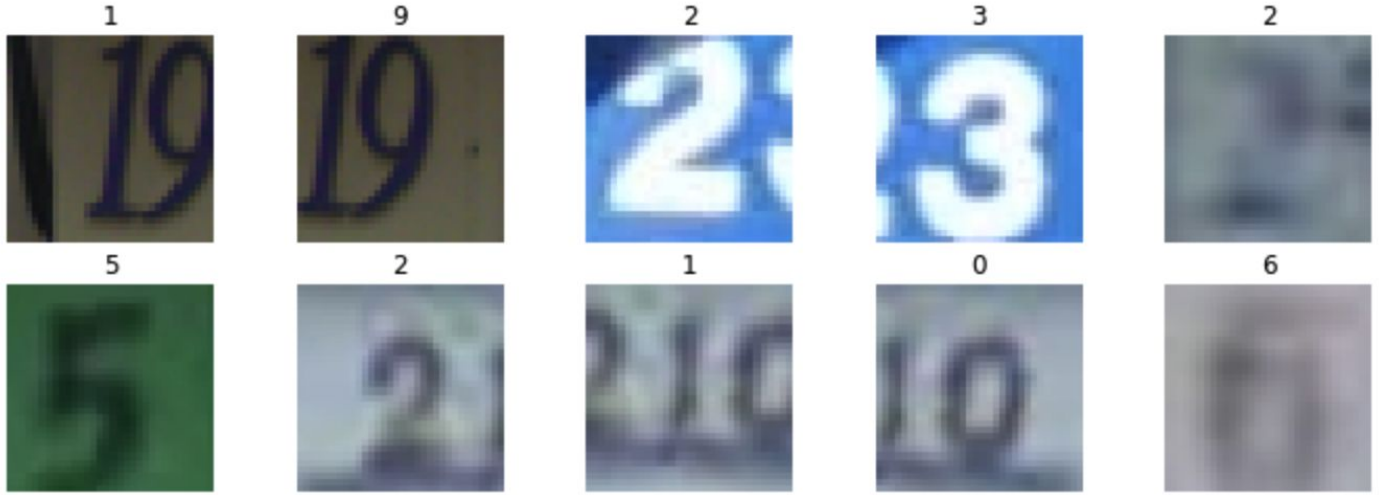
SVHN Dataset

Data Preprocessing

- Normalize the values of the image array so that they are between 0 and 1 for faster computation
- Transpose the numpy arrays so that the index for selecting images comes first
- Convert the label array to binary class matrices from integers

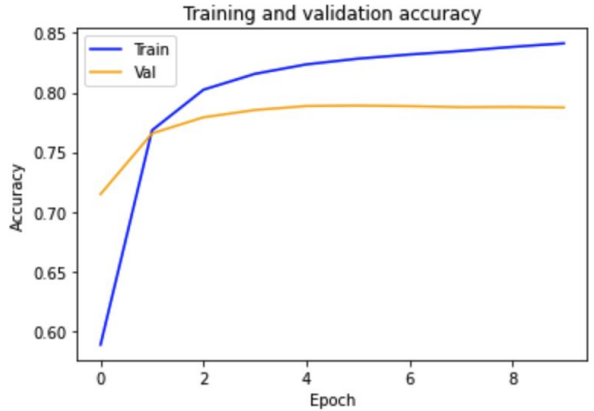
```
Shapes after tranposing the data:
X_train shape: (73257, 32, 32, 3)
train_label shape: (73257,)
X_test shape: (26032, 32, 32, 3)
test_label shape: (26032,)

Number of classes: 10
```

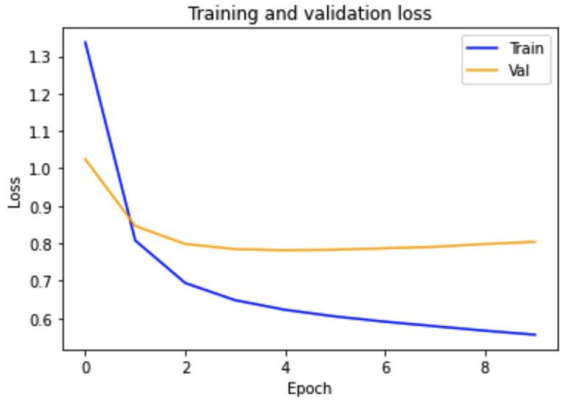
SHVN Dataset Preview



Simple CNN Model

```
model = models.Sequential()
model.add(layers.Conv2D(16, (3, 3), activation='relu',
                        input_shape = (32,32,3)))
model.add(layers.Flatten())
model.add(layers.Dense(10, activation='softmax'))
```



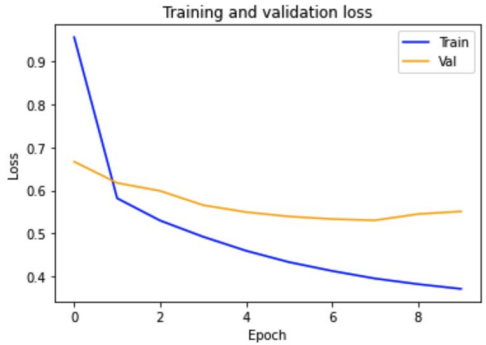


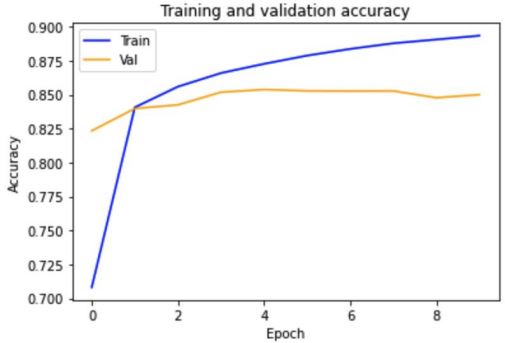

```
814/814 [==============================] - 3s 3ms/step - loss: 0.8036 - accuracy: 0.7875
```

Improved Model with More Layers

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_1 (Conv2D)	(None, 16, 16, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 10)	40970

Total params: 60,362
Trainable params: 60,362
Non-trainable params: 0






```
814/814 [==============================] - 4s 5ms/step - loss: 0.5509 - accuracy: 0.8500
```

Hyperparameter Optimization using KerasTuner

```
Best val_accuracy So Far: 0.9223647713661194
Total elapsed time: 02h 35m 42s
INFO:tensorflow:Oracle triggered exit
Results summary
Results in ./untitled_project
Showing 10 best trials
<keras_tuner.engine.objective.Objective object at 0x7fd46ef59f50>
Trial summary
Hyperparameters:
conv1_filter: 32
conv2_filter: 112
conv3_filter: 512
dense1: 384
dense2: 96
learning_rate: 0.0001
Score: 0.9223647713661194
```

```
814/814 [==============================] - 4s 4ms/step - loss: 0.3362 - accuracy: 0.9224
```

Results


pred=5 | true=5
pred=2 | true=2
pred=1 | true=1
pred=0 | true=0
pred=6 | true=6
pred=1 | true=1
pred=9 | true=9
pred=1 | true=1
pred=1 | true=1
pred=8 | true=8

Conclusion

The goal of this project was to create a CNN model that classifies Street House View Number images into their corresponding categories.

We started with a simple CNN model with one layer, which yielded 79% test accuracy. Then we added some more layers to our base model, gave 85% test accuracy. Finally, we added more layers to the model and ran the Kerastuner function in order to find the best hyperparameters for the model with 25 trials. The best model with the optimized hyperparameter gave us the highest test accuracy of 92%.

Further Improvements

In order to further improve the model, there are a few possible steps that we could try:

1. Data Augmentation
2. Test with more hyperparameters and more layers
3. Transfer learning with pre-trained models