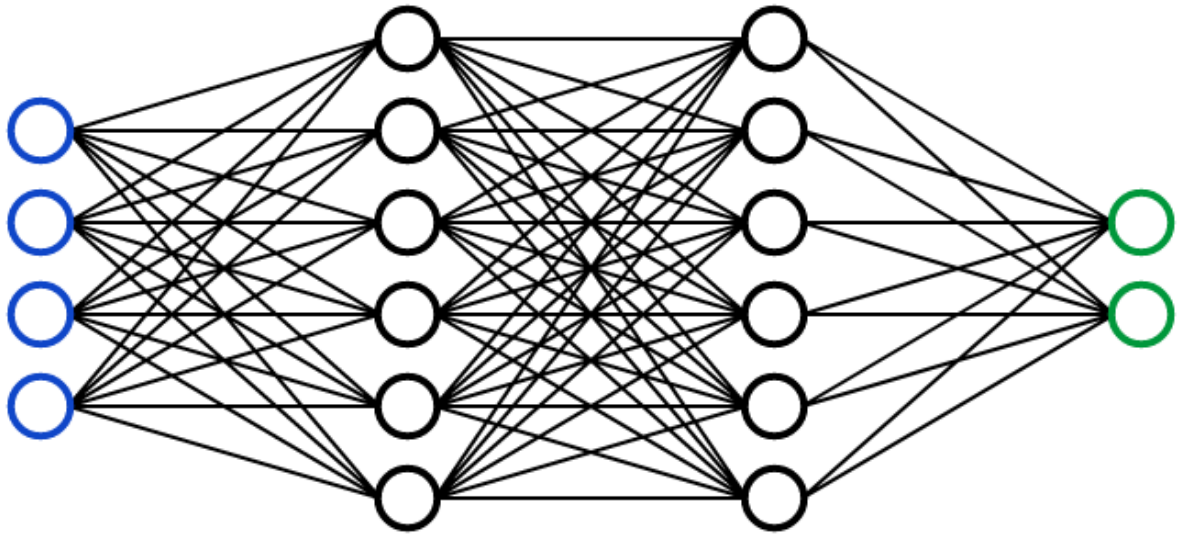
Speed Limit Identifier

Salman Khan

Problem Addressed

Suggested Solution



Dataset

German Traffic Sign Recognition Benchmark (GTSRB) is a labelled dataset with >40 classes and over 50,000 images in total

I picked 7 classes which correspond to speed limits 30, 50, 60, 70, 80, 100, and 120 Km/h

Splitted each class into two groups:

-> 90% for training

-> 10% for validation

Dataset

An example for each of the picked classes:

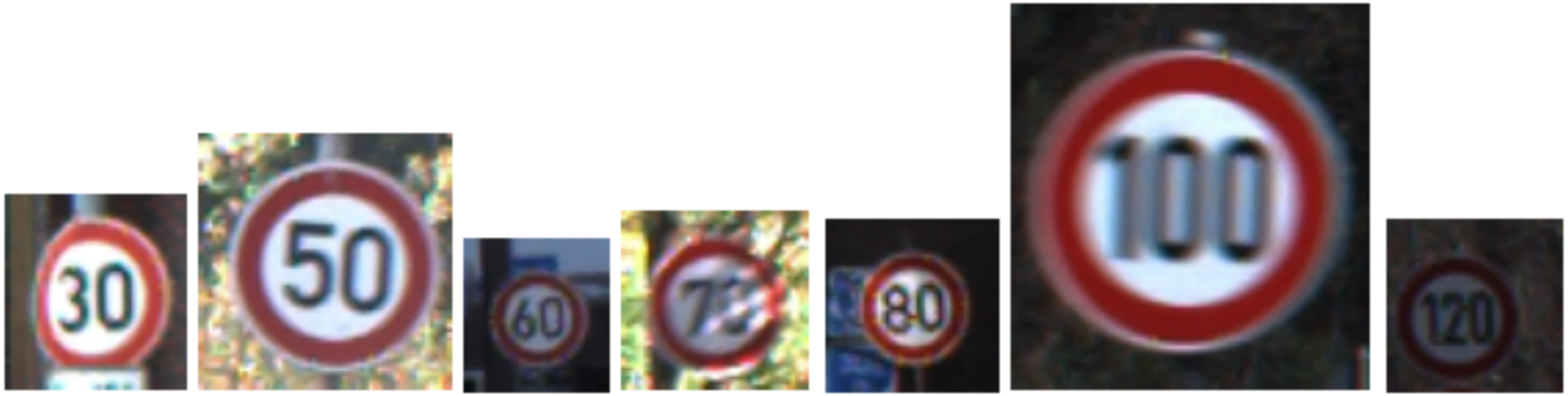
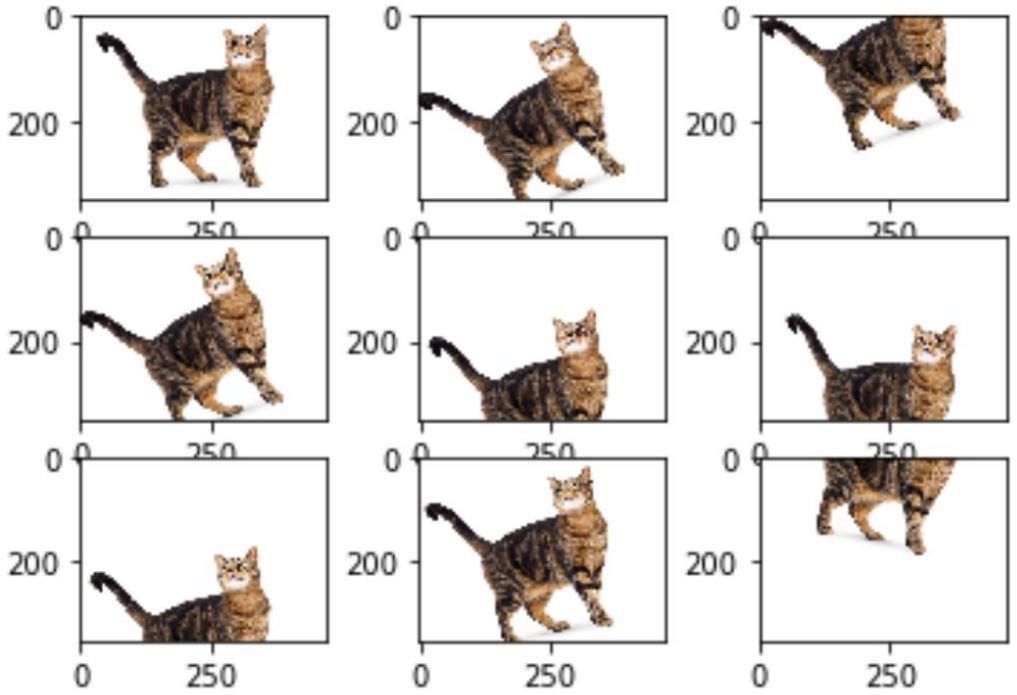


Image Augmentation



Convolutions / Filters

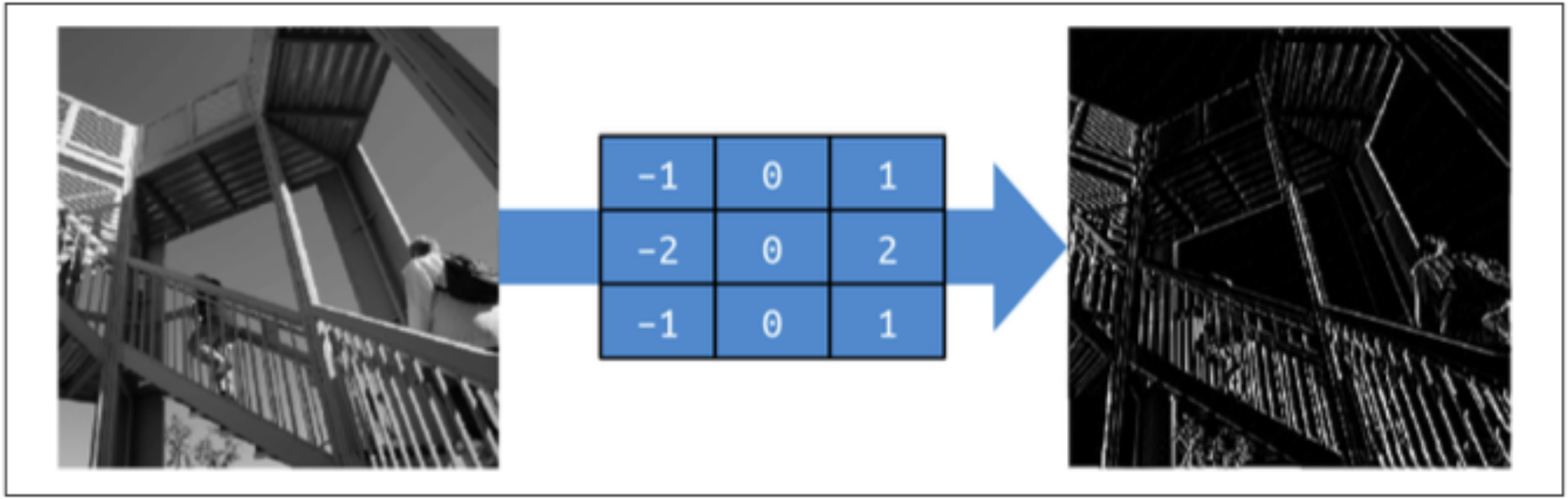


Figure 3-2. Using a filter to get vertical lines

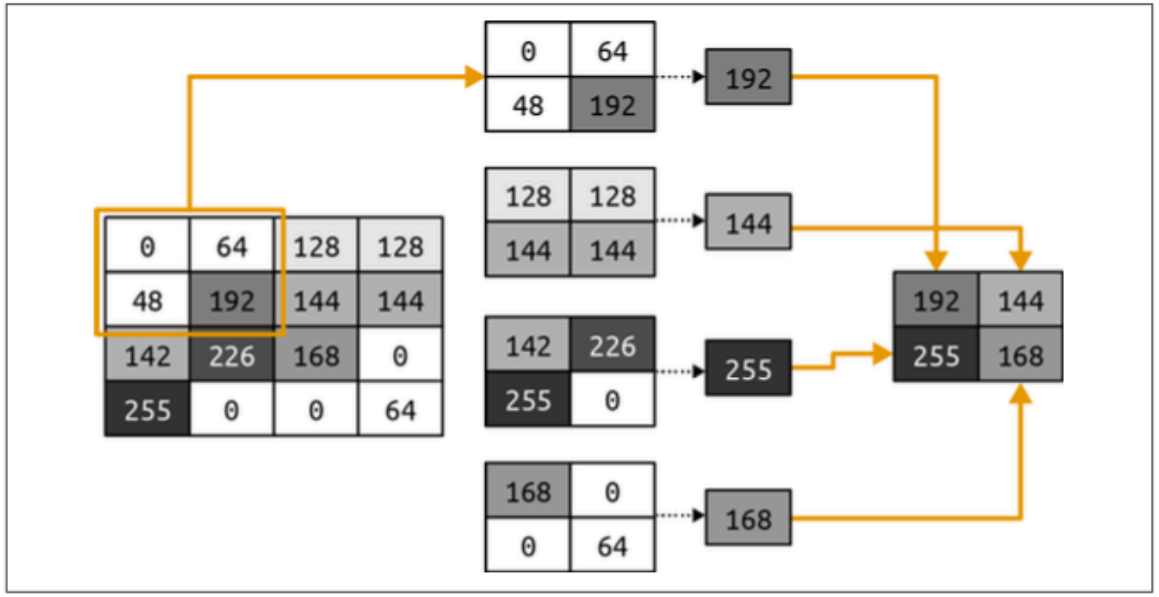


Figure 3-4. Demonstrating max pooling

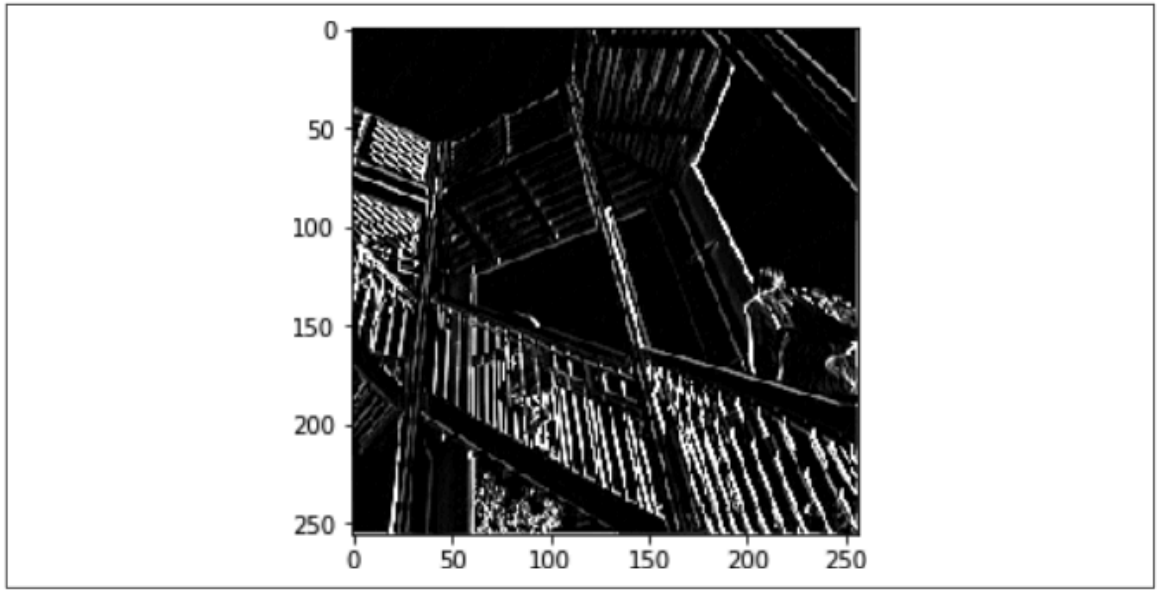


Figure 3-5. Ascent after vertical filter and max pooling

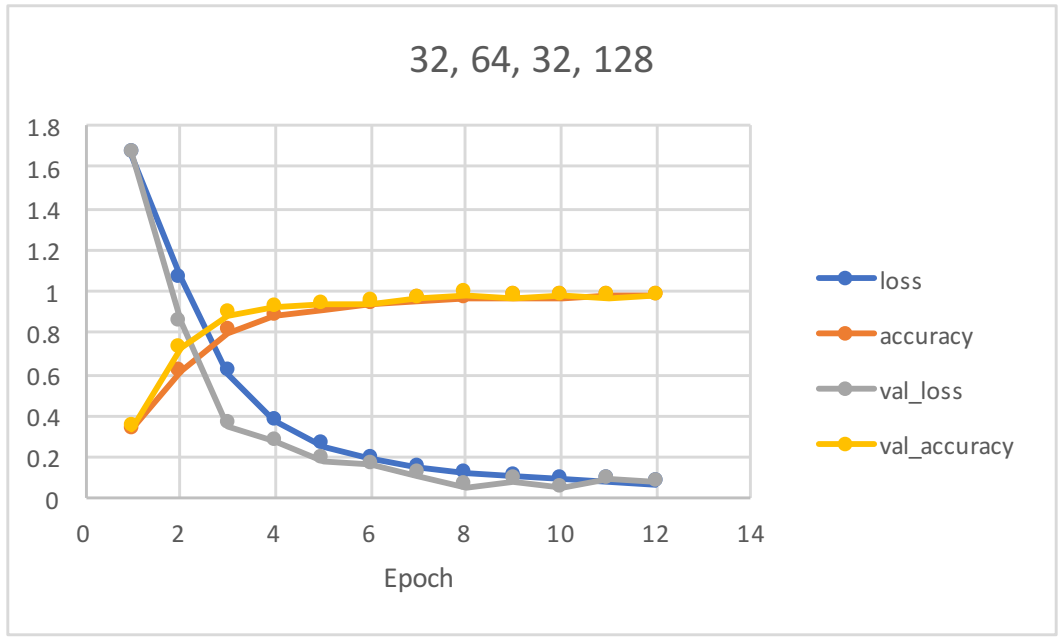
Max Pooling

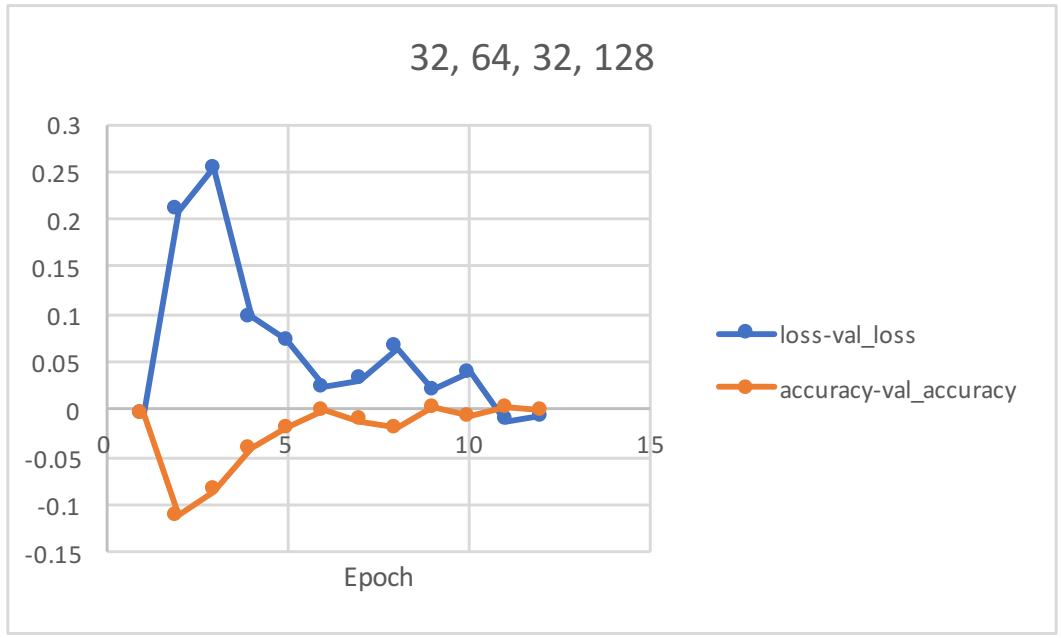
```
model_structure = tf.keras.models.Sequential()


model_structure.add(

tf.keras.layers.Conv2D(hp.Choice('Conv2D_1_filters', values=[16, 32, 64]), (3, 3),
activation='relu', input_shape=(62, 62, 3))

)


model_structure.add(tf.keras.layers.MaxPooling2D(2, 2))


model_structure.add(

tf.keras.layers.Conv2D(hp.Choice('Conv2D_2_filters', values=[16, 32, 64]), (3,
3),activation='relu')

)


model_structure.add(tf.keras.layers.MaxPooling2D(2, 2))
```

RandomSearch / Structure

```
conv3_filters = hp.Choice('Conv2D_3_filters', values=[0, 16, 32, 64])

if conv3_filters != 0:
    model_structure.add(tf.keras.layers.Conv2D(conv3_filters, (3, 3), activation='relu'))

    model_structure.add(tf.keras.layers.MaxPooling2D(2, 2))



model_structure.add(tf.keras.layers.Flatten())

model_structure.add(
tf.keras.layers.Dense(hp.Choice('dense_units', values=[64, 128, 256]), activation='relu')
)

model_structure.add(tf.keras.layers.Dense(7, activation='softmax'))


model_structure.compile(
loss='categorical_crossentropy',
optimizer=tf.keras.optimizers.legacy.RMSprop(),
metrics=['accuracy'])
```

RandomSearch / Structure

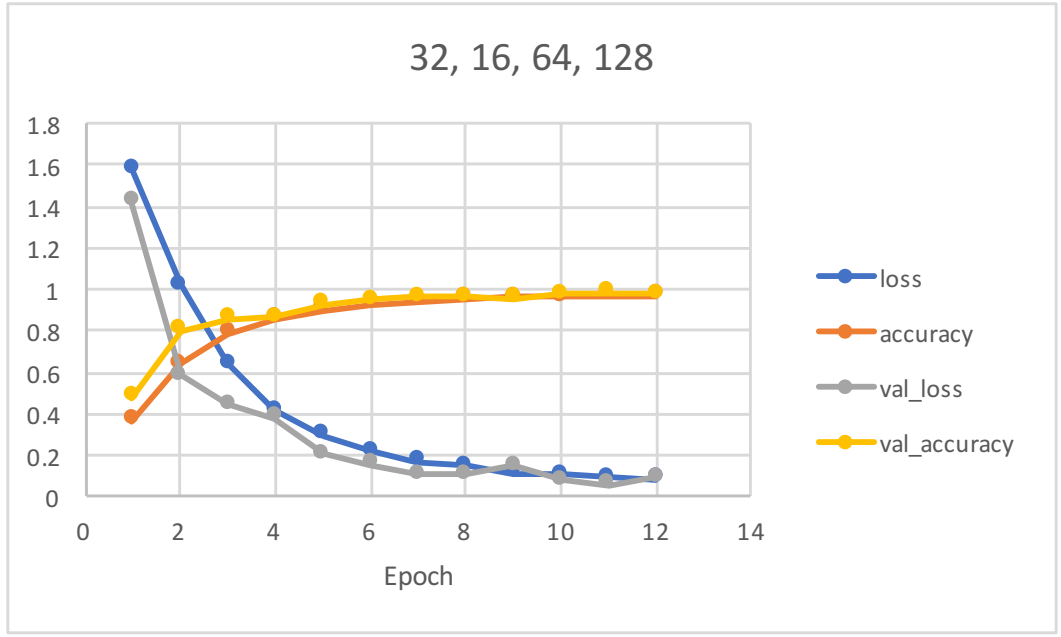


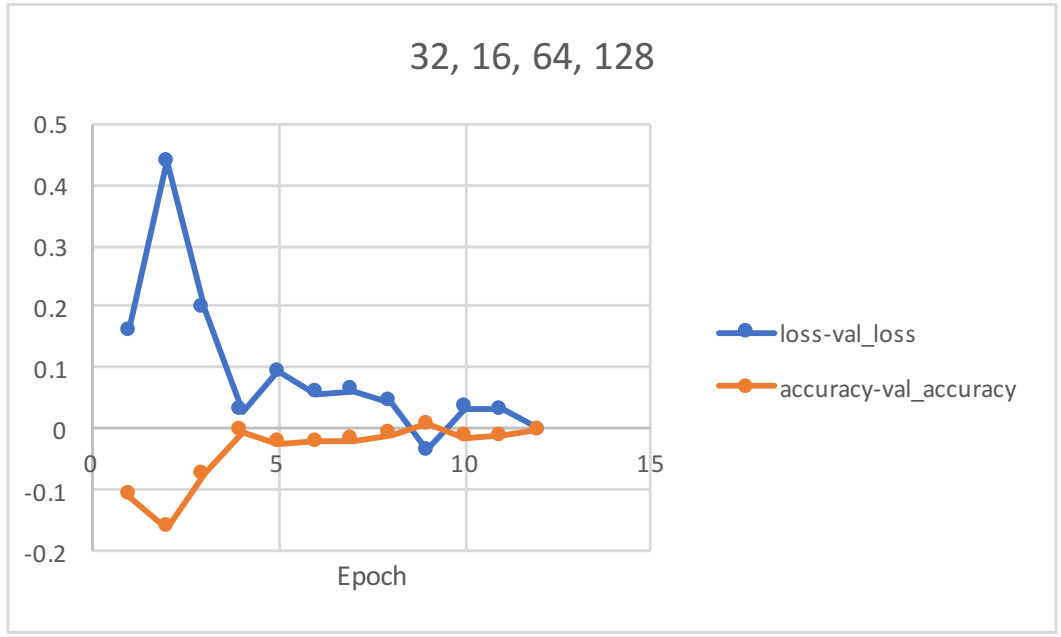


Training Results

(1/2) Quickest to reach desired criteria

In 12 epochs

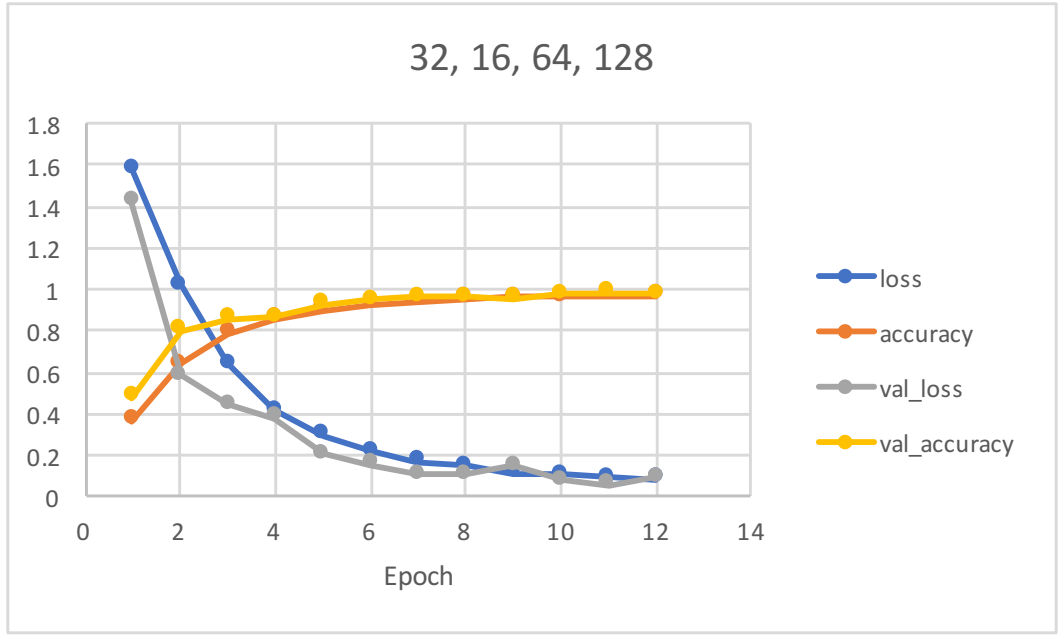


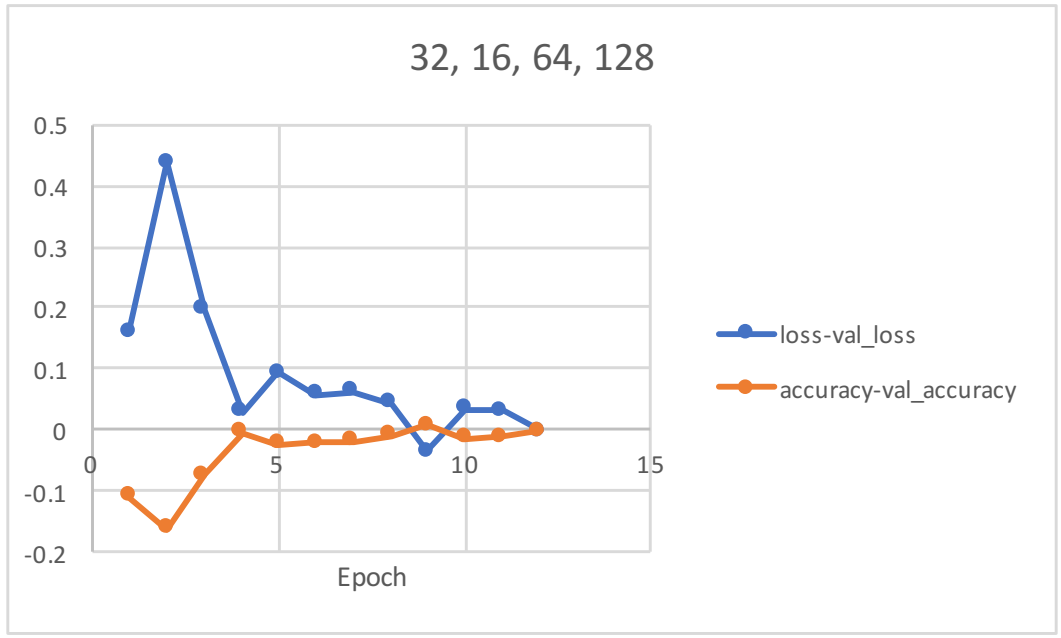


Training Results

(2/2) Quickest to reach desired criteria

In 12 epochs

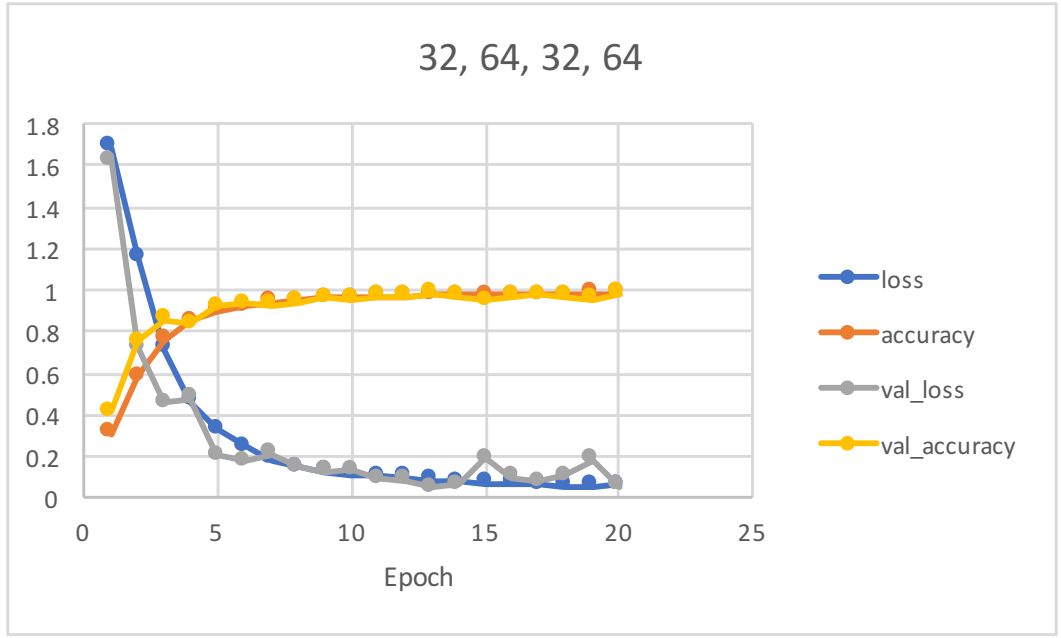


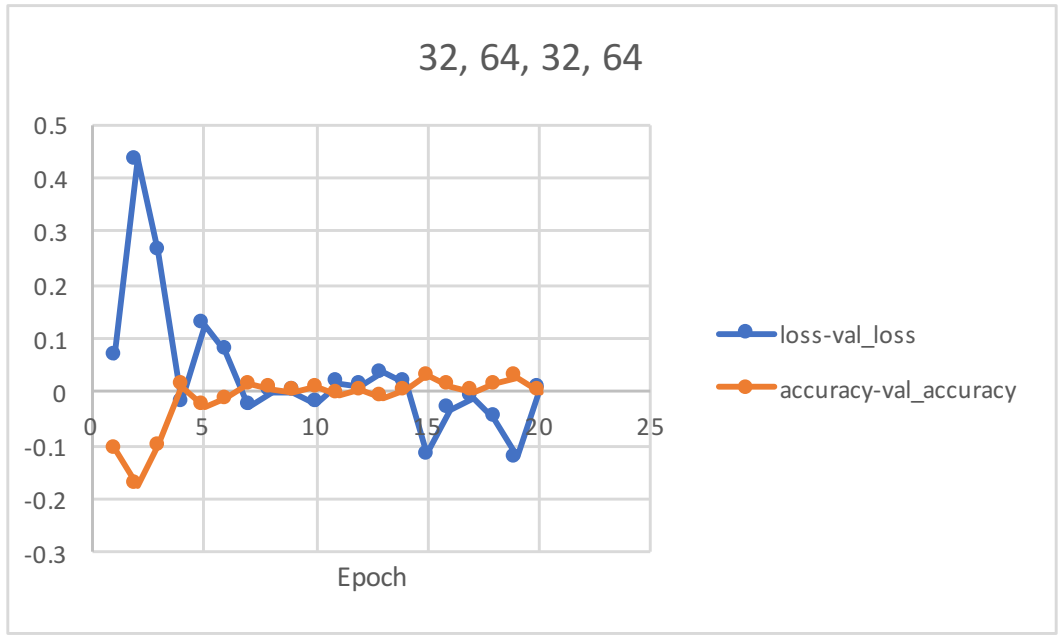


Training Results

Lowest |train_loss – val_loss|

0.0019

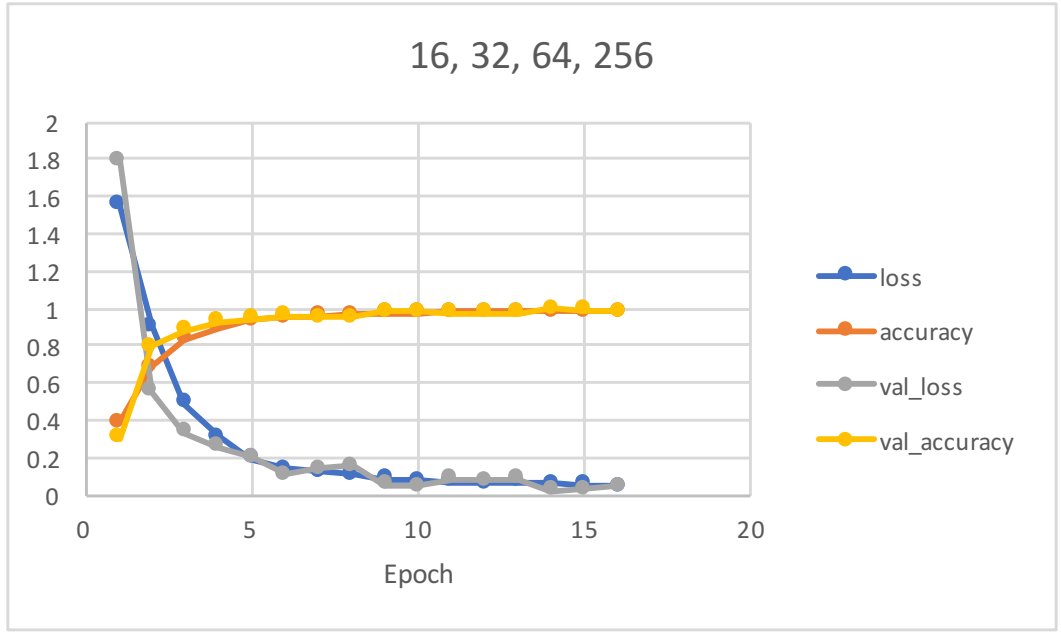


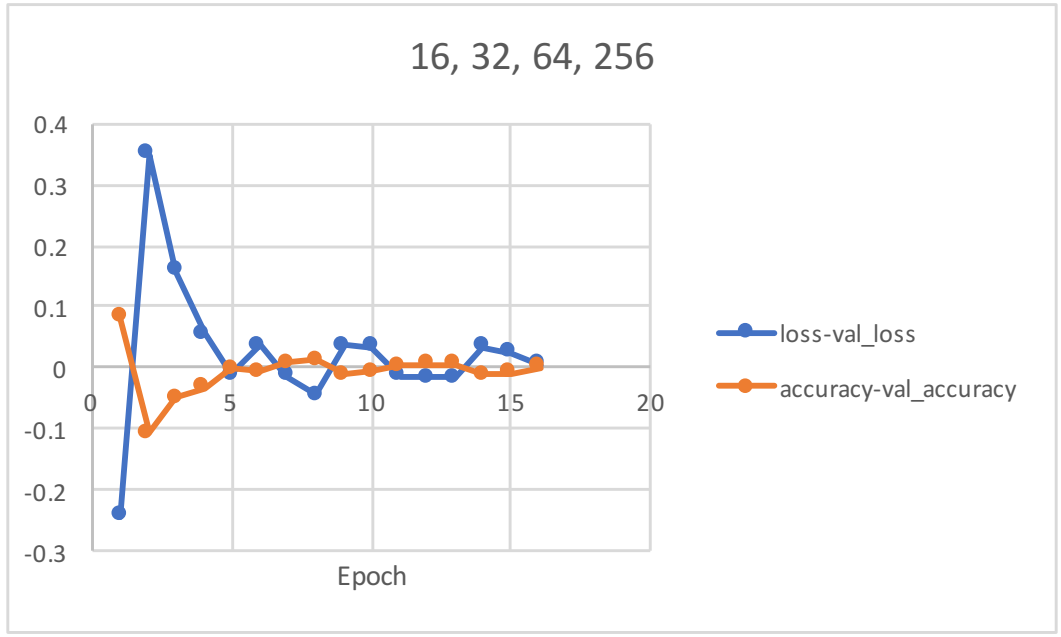


Training Results

Lowest |train_accuracy – val_accuracy|

0.03%





Training Results

Best epoch to epoch convergence

CHOSEN

References

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2019, May 10). *German Traffic Sign Recognition Benchmark GTSRB*. Electronic Research Data Archive. Retrieved December 4, 2022, from https://sid.erda.dk/public/archives/daaeac0d7ce1152aea9b61d9f1e19370/published-archive.html

Moroney, L. (2020). Going Beyond the Basics: Detecting Features in Images. In *AI and Machine Learning for Coders: A programmer's Guide to Artificial Intelligence* (pp. 33–66). O'Reilly.

Moroney, L. (2020). Exploring Hyperparameter Tuning with Keras Tuner. In *AI and Machine Learning for Coders: A programmer's Guide to Artificial Intelligence* (pp. 185–188). O'Reilly.

References

Moroney, L. (2020). Stopping Training. In *AI and Machine Learning for Coders: A programmer's Guide to Artificial Intelligence* (pp. 30–32). O'Reilly.

Silence-tensorflow. PyPI. (n.d.). Retrieved November 27, 2022, from https://pypi.org/project/silence-tensorflow/