

Model 1

Mean Loss: 0.668175
Mean Validation Loss: 0.695348
Batch Size: 32

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))

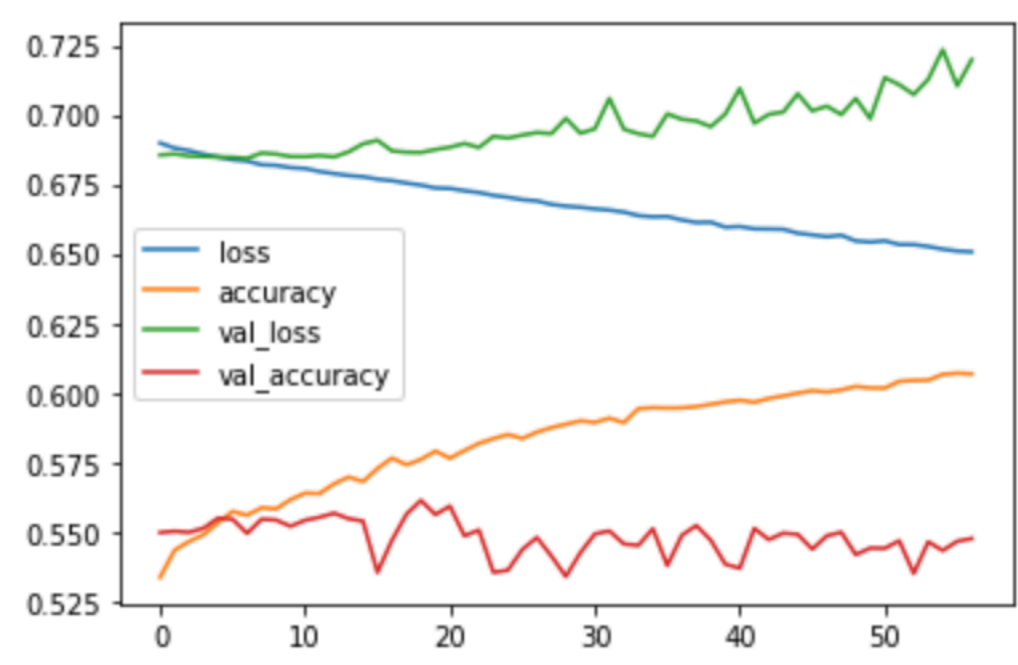
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 2

Mean Loss: 0.664054
Mean Validation Loss: 0.69857
Batch Size: 32

```
model = Sequential()

model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))

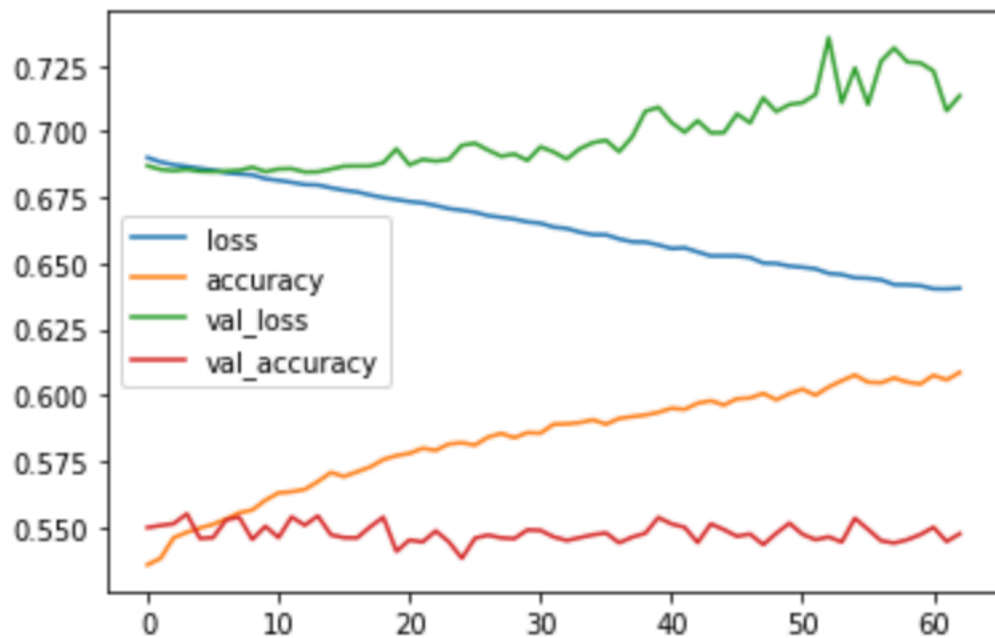
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 3

Mean Loss: 0.620608
Mean Validation Loss: 0.762239
Batch Size: 16

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))

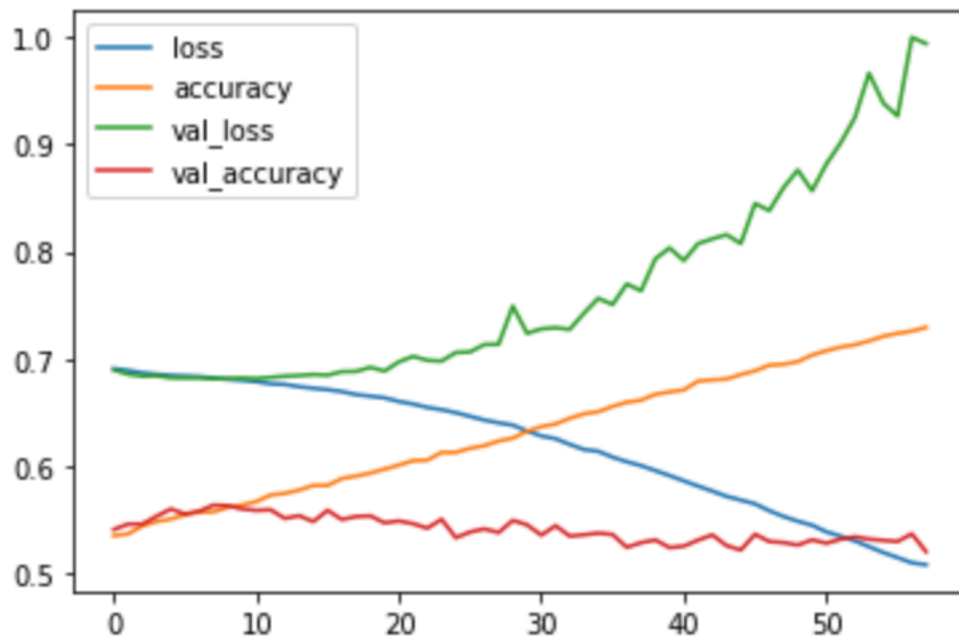
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 4

Mean Loss: 0.673568

Mean Validation Loss: 0.682452

Batch Size: 128

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))

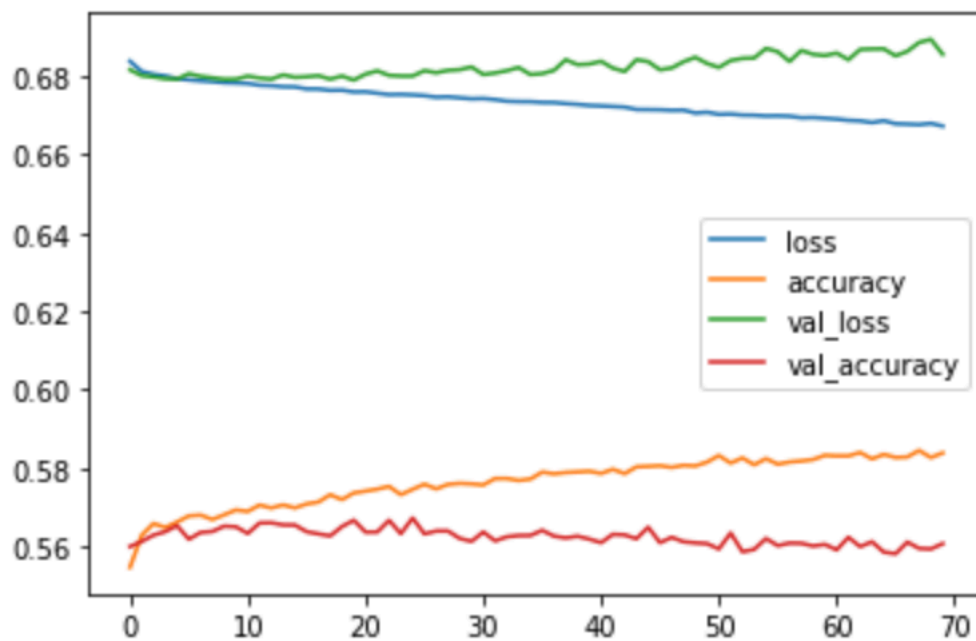
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 5

Mean Loss: 0.684353

Mean Validation Loss: 0.683085

Batch Size: 128

```
model = Sequential()

model.add(Conv2D(filters=8, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))
model.add(Dropout(0.5))

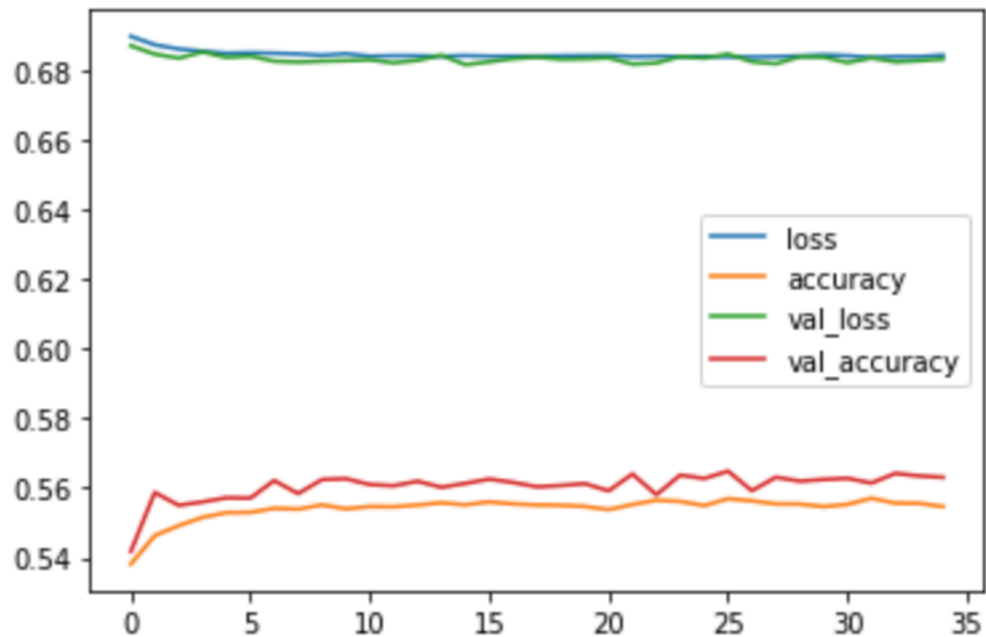
model.add(Conv2D(filters=16, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 6

Mean Loss: 0.68361
Mean Validation Loss: 0.682252
Batch Size: 512

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape, activation='relu'))
model.add(MaxPool2D(pool_size=(3, 3)))
model.add(Dropout(0.5))

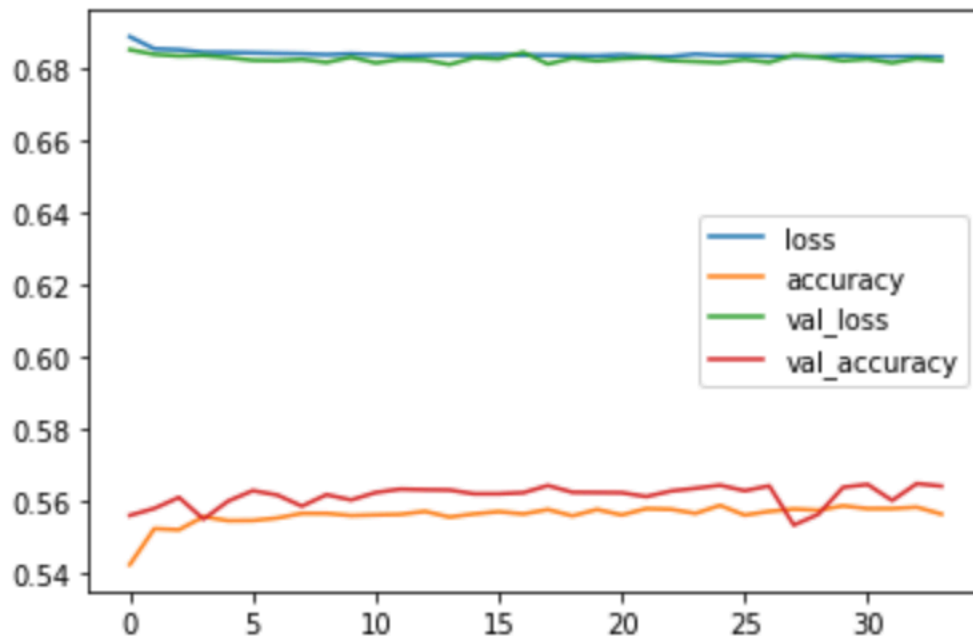
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 7

Mean Loss: 0.681508
Mean Validation Loss: 0.68063
Batch Size: 128

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape, activation='tanh'))
model.add(MaxPool2D(pool_size=(3, 3)))
model.add(Dropout(0.2))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.2))

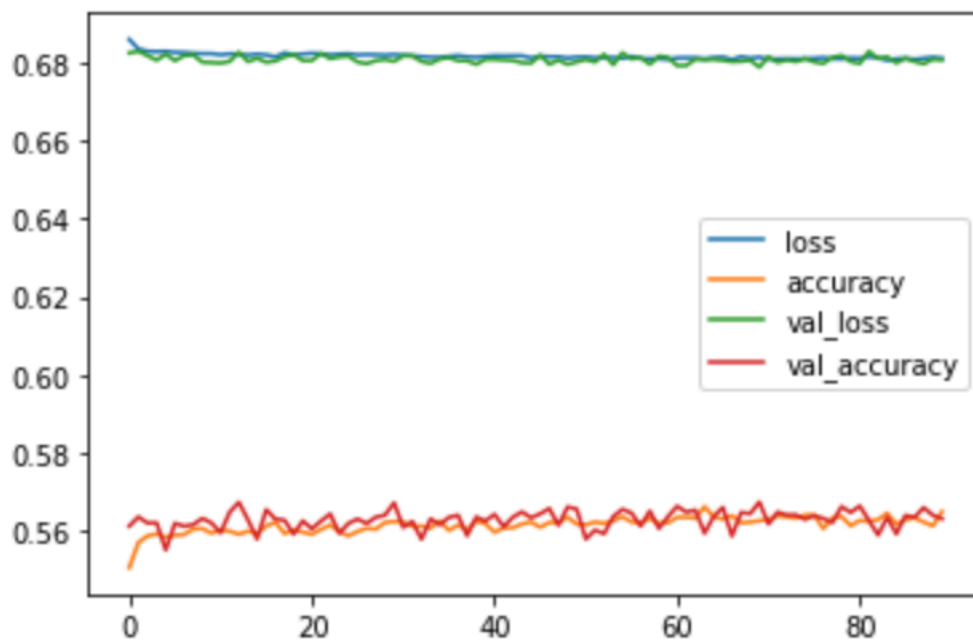
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 8

Mean Loss: 0.678727
Mean Validation Loss: 0.679328
Batch Size: 512

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(1, 1), input_shape=image_shape, activation='tanh'))
model.add(MaxPool2D(pool_size=(1, 1)))

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=image_shape,
                  activation='tanh', padding="same"))
model.add(MaxPool2D(pool_size=(3, 3)))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))

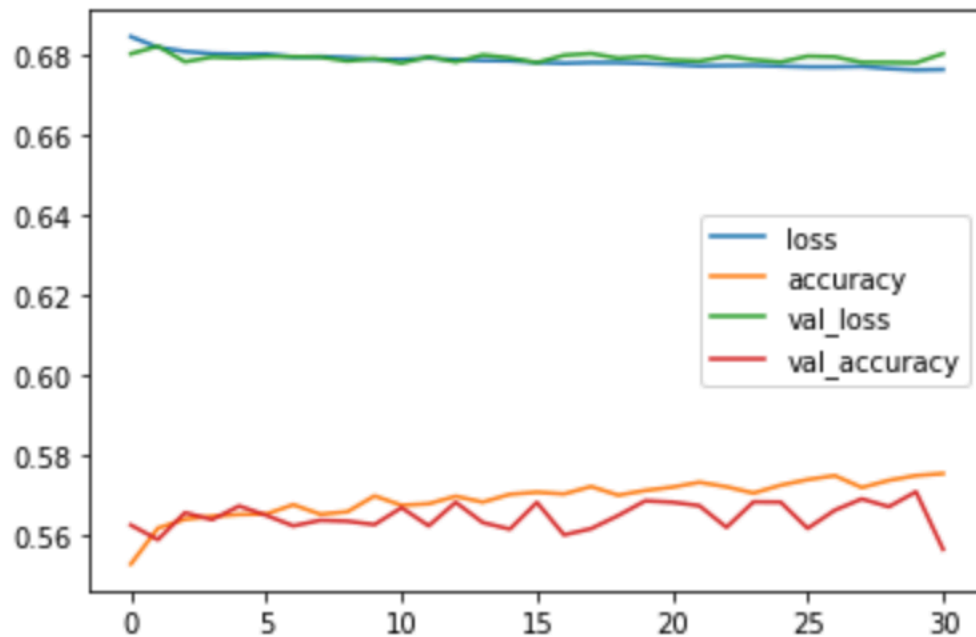
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```



Model 9

Mean Loss: 0.683104
Mean Validation Loss: 0.682327
Batch Size: 16

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(1, 1),
                  input_shape=image_shape, activation='tanh'))
model.add(MaxPool2D(pool_size=(1, 1)))

model.add(Conv2D(filters=32, kernel_size=(3, 3),
                  input_shape=image_shape, activation='tanh'))
model.add(MaxPool2D(pool_size=(3, 3)))
model.add(Dropout(0.5))

model.add(Conv2D(filters=64, kernel_size=(3, 3),
                  activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(filters=128, kernel_size=(3, 3),
                  activation='relu', padding="same"))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss=loss_param, optimizer=optimizer_param,
              metrics=[metric])
```

