

## ▼ CatDog Classification with CNN

```
# To mount the drive
from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense,Dropout,BatchNormalization
from tensorflow.keras.regularizers import l2

classifier = Sequential()

# Convolution
classifier.add(Conv2D(32, 5, 5, input_shape = (256, 256, 3), activation = 'relu',kernel_regularizer=l2(l2=0.01)))
# classifier.add(BatchNormalization())

# Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
classifier.add(Dropout(0.4))

# Adding a second convolutional layer
classifier.add(Conv2D(64, 5, 5, activation = 'relu',kernel_regularizer=l2(l2=0.01)))
# classifier.add(BatchNormalization())
classifier.add(MaxPooling2D(pool_size = (2, 2)))
classifier.add(Dropout(0.4))

# Flattening
classifier.add(Flatten())

# Full connection
classifier.add(Dense(128, activation = 'relu'))
classifier.add(Dropout(0.3))
classifier.add(Dense(1, activation = 'sigmoid'))

classifier.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 51, 51, 32)	2432
max_pooling2d (MaxPooling2D)	(None, 25, 25, 32)	0
dropout (Dropout)	(None, 25, 25, 32)	0
conv2d_1 (Conv2D)	(None, 5, 5, 64)	51264
max_pooling2d_1 (MaxPooling2D)	(None, 2, 2, 64)	0
dropout_1 (Dropout)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dropout_2 (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33
dense_2 (Dense)	(None, 128)	256
dropout_3 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

=====

Total params: 62,338

Trainable params: 62,338  
Non-trainable params: 0

---

```
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

```
# To generate more images
```

```
from keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2, # Rotate img anticlockwise
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
training_set = train_datagen.flow_from_directory('/content/drive/MyDrive/CNN_Project/data/train',
                                                target_size = (256, 256),
                                                batch_size = 16,
                                                class_mode = 'binary')
```

Found 40 images belonging to 2 classes.

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

```
test_set = test_datagen.flow_from_directory('/content/drive/MyDrive/CNN_Project/data/test',
                                           target_size = (256, 256),
                                           batch_size = 16,
                                           class_mode = 'binary')
```

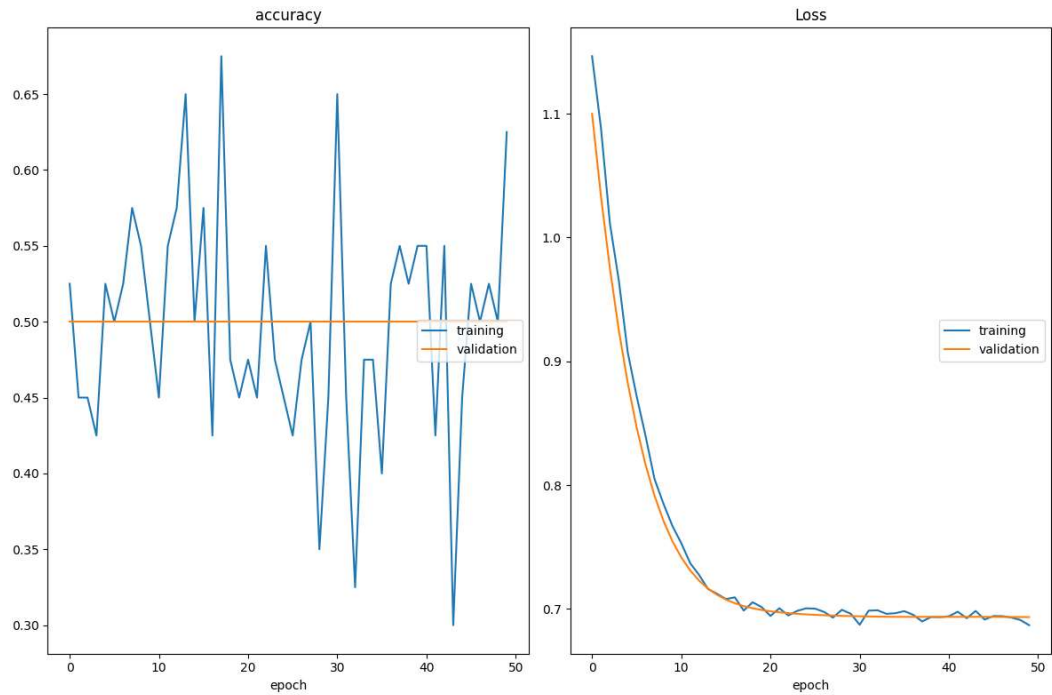
Found 20 images belonging to 2 classes.

```
! pip install livelossplot
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting livelossplot
  Downloading livelossplot-0.5.5-py3-none-any.whl (22 kB)
Requirement already satisfied: bokeh in /usr/local/lib/python3.10/dist-packages (from livelossplot) (2.4.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from livelossplot) (3.7.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (6.2)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (3.1.2)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (6.0)
Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (23.1)
Requirement already satisfied: typing-extensions>=3.10.0 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (4.5)
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (1.22.4)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from bokeh->livelossplot) (8.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (3.0.9)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (4.39.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->livelossplot) (2.8)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->bokeh->livelossplot) (2.0.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->livelossplot) (1.16.0)
Installing collected packages: livelossplot
Successfully installed livelossplot-0.5.5
```

```
from livelossplot import PlotLossesKerasTF
```

```
classifier.fit(training_set, epochs=50, validation_data = test_set, callbacks=[PlotLossesKerasTF()])
```

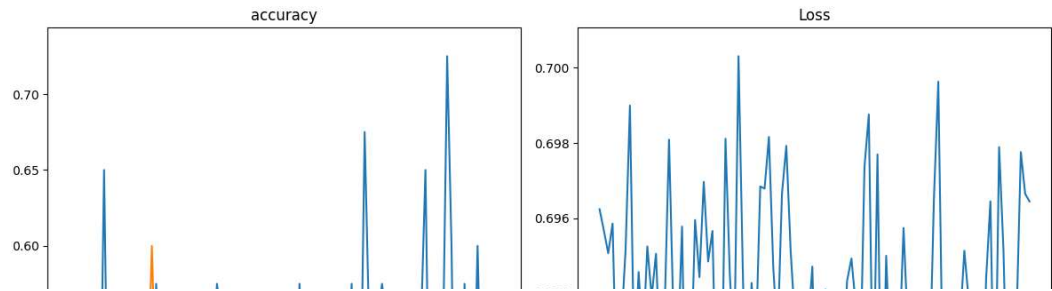


```
accuracy
training      (min:  0.300, max:  0.675, cur:  0.625)
validation    (min:  0.500, max:  0.500, cur:  0.500)

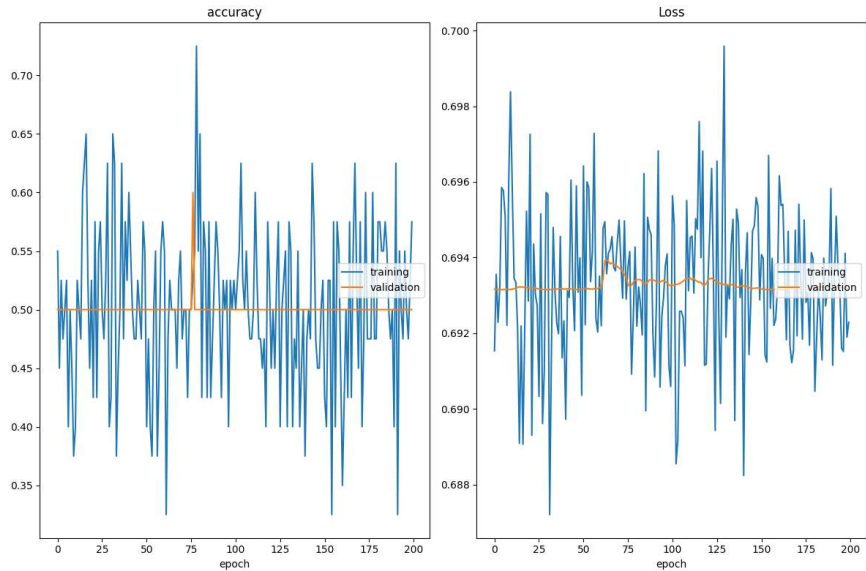
Loss
training      (min:  0.687, max:  1.146, cur:  0.687)
validation    (min:  0.693, max:  1.100, cur:  0.693)

3/3 [=====] - 1s 583ms/step - loss: 0.6868 - accuracy: 0.6250 - val_loss: 0.6930
<keras.callbacks.History at 0x7f561ebaec80>
```

```
classifier.fit(training_set,epochs=100,validation_data = test_set,callbacks=[PlotLossesKerasTF()])
```



```
classifier.fit(training_set,epochs=200,validation_data = test_set,callbacks=[PlotLossesKerasTF()])
```



```
accuracy
training      (min: 0.325, max: 0.725, cur: 0.575)
validation    (min: 0.500, max: 0.600, cur: 0.500)

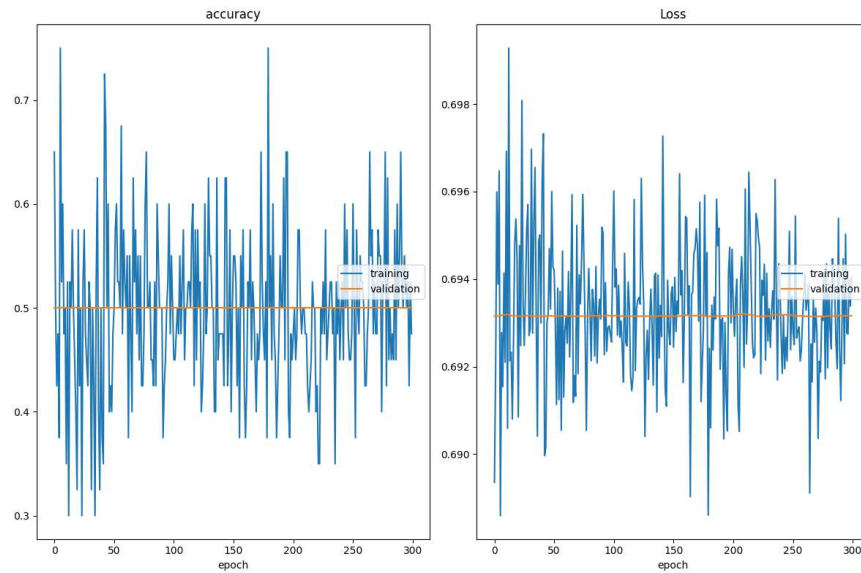
Loss
training      (min: 0.687, max: 0.700, cur: 0.692)
validation    (min: 0.693, max: 0.694, cur: 0.693)

3/3 [=====] - 2s 754ms/step - loss: 0.6923 - accuracy: 0.575
<keras.callbacks.History at 0x7f54eacbd360>
```

Double-click (or enter) to edit

```
classifier.fit(training_set,epochs=300,validation_data = test_set,callbacks=[PlotLossesKerasTF()])
```





```

accuracy
  training      (min: 0.300, max: 0.750, cur: 0.475)
  validation    (min: 0.500, max: 0.500, cur: 0.500)
Loss
  training      (min: 0.689, max: 0.699, cur: 0.694)
  validation    (min: 0.693, max: 0.693, cur: 0.693)
3/3 [=====] - 1s 492ms/step - loss: 0.6940 - accuracy: 0.475
<keras.callbacks.History at 0x7f54ea9461a0>

```

```

import numpy as np
# from keras.preprocessing import image
from tensorflow.keras.preprocessing import image

test_image = image.load_img('/content/drive/MyDrive/CNN_Project/data/train/dogs/8.jpg',target_size=(256,256))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image,axis=0)
result = classifier.predict(test_image)

if result[0][0]>0.5:
    prediction= 'dogs'
    print('Result is',result[0][0])
else:
    prediction = 'cats'
    print('Result is',result[0][0])

print(prediction)

1/1 [=====] - 0s 34ms/step
Result is 0.50249976
dogs

```

```

import numpy as np
# from keras.preprocessing import image
from tensorflow.keras.preprocessing import image

test_image = image.load_img('/content/drive/MyDrive/CNN_Project/data/train/cats/11.jpg',target_size=(256,256))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image,axis=0)
result = classifier.predict(test_image)
training_set.class_indices

if result[0][0]>0.5:
    prediction= 'dogs'
    print('Result is',result[0][0])
else:
    prediction = 'cats'
    print('Result is',result[0][0])

print(prediction)

```

```
1/1 [=====] - 0s 27ms/step
Result is 0.5024998
dogs
```

✓ 0s completed at 13:43 ● ✕