In [47]:

```
import cv2
from cv2 import IMREAD_COLOR,IMREAD_UNCHANGED
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from scipy.ndimage import variance
from skimage import io
from  skimage.color import rgb2gray
from skimage.filters import laplace,sobel,roberts
from skimage.transform import resize
%matplotlib inline
import tensorflow as tf
```

In [48]:

```
import cv2
```

In [49]:

```
import os
Directory = '../input/newset'
Categories = ['New folder (2)']
data = []
for category in Categories:
    folder = os.path.join(Directory,category)
    label = Categories.index(category)
    for img in os.listdir(folder):
        img_path = os.path.join(folder,img)
        img_arr = cv2.imread(img_path)
        img_gray = cv2.cvtColor(img_arr,cv2.COLOR_BGR2GRAY)

        laplacian_var = cv2.Laplacian(img_gray,cv2.CV_16S) #64F
        data.append([laplacian_var,label])
```

In [50]:

```
import random
```

In [51]:

```
random.shuffle(data)
```

In [52]:

```
X = []
y = []

for feature , label in data:
    X.append(feature)
    y.append(label)
```

In [53]:

```
import numpy as np
X = np.array(X)
y = np.array(y)
```

In [54]:

```
X = X/255
```

In [56]:

```
X.shape
```

Out[56]:

```
(20, 4000, 6000)
```

In [55]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Flatten,Dense
```

In [57]:

```
model = Sequential()
model.add(Conv2D(64,(4,4),activation='relu'))
model.add(MaxPooling2D(2,2))

model.add(Conv2D(128,(4,4),activation='relu'))
model.add(MaxPooling2D(2,2))

model.add(Flatten())

model.add(Dense(128,input_shape=X.shape[0:], activation='relu'))
model.add(Dense(1, activation='softmax'))
```

In [58]:

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

In [59]:

```
model.fit(X,y,epochs=5,validation_split=0.1)
```

```
Epoch 1/5


---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-59-885523f3170b> in <module>
----> 1 model.fit(X,y,epochs=5,validation_split=0.1)

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py in fit(self, x,
y, batch_size, epochs, verbose, callbacks, validation_split, validation_data, shuffle,
class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_steps,
validation_batch_size, validation_freq, max_queue_size, workers, use_multiprocessing)
   1098                 _r=1):
   1099               callbacks.on_train_batch_begin(step)
-> 1100               tmp_logs = self.train_function(iterator)
   1101               if data_handler.should_sync:
   1102                 context.async_wait()

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in __call__(self, *
args, **kwds)
    826     tracing_count = self.experimental_get_tracing_count()
    827     with trace.Trace(self._name) as tm:
--> 828       result = self._call(*args, **kwds)
    829       compiler = "xla" if self._experimental_compile else "nonXla"
    830       new_tracing_count = self.experimental_get_tracing_count()

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in _call(self,
*args, **kwds)
    869       # This is the first call of __call__, so we have to initialize.
    870       initializers = []
--> 871       self._initialize(args, kwds, add_initializers_to=initializers)
    872     finally:
    873       # At this point we know that the initialization is complete (or less

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in
_initialize(self, args, kwds, add_initializers_to)
    724     self._concrete_stateful_fn = (
    725         self._stateful_fn._get_concrete_function_internal_garbage_collected(  # pylint: dis
able=protected-access
```

```
able-protected-access
--> 726             *args, **kwds))
    727
    728     def invalid_creator_scope(*unused_args, **unused_kwds):

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_get_concrete_function_internal_garbage_collected(self, *args, **kwargs)
   2967       args, kwargs = None, None
   2968     with self._lock:
-> 2969       graph_function, _ = self._maybe_define_function(args, kwargs)
   2970     return graph_function
   2971

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_maybe_define_function(self, args, kwargs)
   3359
   3360           self._function_cache.missed.add(call_context_key)
-> 3361           graph_function = self._create_graph_function(args, kwargs)
   3362           self._function_cache.primary[cache_key] = graph_function
   3363

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_create_graph_function(self, args, kwargs, override_flat_arg_shapes)
   3204             arg_names=arg_names,
   3205             override_flat_arg_shapes=override_flat_arg_shapes,
-> 3206             capture_by_value=self._capture_by_value),
   3207         self._function_attributes,
   3208         function_spec=self.function_spec,

/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in
func_graph_from_py_func(name, python_func, args, kwargs, signature, func_graph, autograph,
autograph_options, add_control_dependencies, arg_names, op_return_value, collections,
capture_by_value, override_flat_arg_shapes)
    988         _, original_func = tf_decorator.unwrap(python_func)
    989
--> 990       func_outputs = python_func(*func_args, **func_kwargs)
    991
    992       # invariant: `func_outputs` contains only Tensors, CompositeTensors,

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in
wrapped_fn(*args, **kwds)
    632             xla_context.Exit()
    633         else:
--> 634           out = weak_wrapped_fn().__wrapped__(*args, **kwds)
    635         return out
    636

/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in wrapper(*args,
**kwargs)
    975           except Exception as e:  # pylint:disable=broad-except
    976             if hasattr(e, "ag_error_metadata"):
--> 977               raise e.ag_error_metadata.to_exception(e)
    978             else:
    979               raise

ValueError: in user code:

    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:805
train_function  *
        return step_function(self, iterator)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:795
step_function  **
        outputs = model.distribute_strategy.run(run_step, args=(data,))
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:1259 run
        return self._extended.call_for_each_replica(fn, args=args, kwargs=kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:2730
call_for_each_replica
        return self._call_for_each_replica(fn, args, kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:3417
_call_for_each_replica
        return fn(*args, **kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:788 run_step
**
        outputs = model.train_step(data)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:754
train_step
        y_pred = self(x, training=True)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/base_layer.py:998
```

```
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/base_layer.py:996
__call__
        input_spec.assert_input_compatibility(self.input_spec, inputs, self.name)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/input_spec.py:239
assert_input_compatibility
        str(tuple(shape)))

    ValueError: Input 0 of layer sequential_4 is incompatible with the layer: : expected
min_ndim=4, found ndim=3. Full shape received: (None, 4000, 6000)
```

In [60]:

```
path = '../input/newset/New folder (2)/'
path1 = '../input/uni-campus-dataset/RGB-img/img/'
```

In [61]:

```
blur_img  = os.listdir(path)


blur_img1 = os.listdir(path1)
```

In [62]:

```
import pandas as pd
def get_data(path,images):
    features = []
    for img in  images:
        feature = []
        gray = cv2.imread(path+img,cv2.COLOR_BGR2GRAY)
        laplacian_var = cv2.Laplacian(gray,cv2.CV_64F).var()
        feature.append(laplacian_var)
        features.append(feature)
    return features
```

In [63]:

```
features = get_data(path,blur_img)
```

In [64]:

```
features1 = get_data(path1,blur_img1)
```

In [65]:

```
features_df = pd.DataFrame(features)
features_df1 = pd.DataFrame(features1)
```

In [88]:

```
print(features_df.shape, features_df1.shape)
```

```
(20, 1) (442, 1)
```

In [90]:

```
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,confusion_matrix,f1_score,classification_report


images = pd.DataFrame()
images = images.append(features_df)
images = images.append(features_df1)


images = np.array(images)
```

In [92]:

```
X,y = train_test_split(images, test_size=0.5)
```

In [94]:

```
model.fit(X,y,epochs=5)
```

```
Epoch 1/5


---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-94-9eed191458d8> in <module>
----> 1 model.fit(X,y,epochs=5)

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py in fit(self, x,
y, batch_size, epochs, verbose, callbacks, validation_split, validation_data, shuffle,
class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_steps,
validation_batch_size, validation_freq, max_queue_size, workers, use_multiprocessing)
   1098                 _r=1):
   1099               callbacks.on_train_batch_begin(step)
-> 1100               tmp_logs = self.train_function(iterator)
   1101               if data_handler.should_sync:
   1102                 context.async_wait()

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in __call__(self, *
args, **kwds)
    826     tracing_count = self.experimental_get_tracing_count()
    827     with trace.Trace(self._name) as tm:
--> 828       result = self._call(*args, **kwds)
    829       compiler = "xla" if self._experimental_compile else "nonXla"
    830       new_tracing_count = self.experimental_get_tracing_count()

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in _call(self,
*args, **kwds)
    860       # In this case we have not created variables on the first call. So we can
    861       # run the first trace but we should fail if variables are created.
--> 862       results = self._stateful_fn(*args, **kwds)
    863       if self._created_variables:
    864         raise ValueError("Creating variables on a non-first call to a function"

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in __call__(self,
*args, **kwargs)
   2939     with self._lock:
   2940       (graph_function,
-> 2941        filtered_flat_args) = self._maybe_define_function(args, kwargs)
   2942     return graph_function._call_flat(
   2943         filtered_flat_args, captured_inputs=graph_function.captured_inputs)  # pylint: disa
ble=protected-access

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_maybe_define_function(self, args, kwargs)
   3356               call_context_key in self._function_cache.missed):
   3357             return self._define_function_with_shape_relaxation(
-> 3358                 args, kwargs, flat_args, filtered_flat_args, cache_key_context)
   3359
   3360           self._function_cache.missed.add(call_context_key)

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_define_function_with_shape_relaxation(self, args, kwargs, flat_args, filtered_flat_args,
cache_key_context)
   3278
   3279     graph_function = self._create_graph_function(
-> 3280         args, kwargs, override_flat_arg_shapes=relaxed_arg_shapes)
   3281     self._function_cache.arg_relaxed[rank_only_cache_key] = graph_function
   3282

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in
_create_graph_function(self, args, kwargs, override_flat_arg_shapes)
   3204             arg_names=arg_names,
   3205             override_flat_arg_shapes=override_flat_arg_shapes,
-> 3206             capture_by_value=self._capture_by_value),
   3207         self._function_attributes,
   3208         function_spec=self.function_spec,
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in
func_graph_from_py_func(name, python_func, args, kwargs, signature, func_graph, autograph,
autograph_options, add_control_dependencies, arg_names, op_return_value, collections,
capture_by_value, override_flat_arg_shapes)
    988         _, original_func = tf_decorator.unwrap(python_func)
    989
--> 990       func_outputs = python_func(*func_args, **func_kwargs)
    991
    992       # invariant: `func_outputs` contains only Tensors, CompositeTensors,

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/def_function.py in
wrapped_fn(*args, **kwds)
    632             xla_context.Exit()
    633         else:
--> 634           out = weak_wrapped_fn().__wrapped__(*args, **kwds)
    635         return out
    636

/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in wrapper(*args,
**kwargs)
    975           except Exception as e:  # pylint:disable=broad-except
    976             if hasattr(e, "ag_error_metadata"):
--> 977               raise e.ag_error_metadata.to_exception(e)
    978             else:
    979               raise

ValueError: in user code:

    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:805
train_function  *
        return step_function(self, iterator)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:795
step_function  **
        outputs = model.distribute_strategy.run(run_step, args=(data,))
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:1259 run
        return self._extended.call_for_each_replica(fn, args=args, kwargs=kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:2730
call_for_each_replica
        return self._call_for_each_replica(fn, args, kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/distribute/distribute_lib.py:3417
_call_for_each_replica
        return fn(*args, **kwargs)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:788 run_step
**
        outputs = model.train_step(data)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:754
train_step
        y_pred = self(x, training=True)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/base_layer.py:998
__call__
        input_spec.assert_input_compatibility(self.input_spec, inputs, self.name)
    /opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/input_spec.py:239
assert_input_compatibility
        str(tuple(shape)))

    ValueError: Input 0 of layer sequential_4 is incompatible with the layer: : expected
min_ndim=4, found ndim=2. Full shape received: (None, 1)
```

In [ ]: