PROJECT BASED LAB REPORT

On

Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure

Submitted in partial fulfilment of the

Requirements for the award of the Degree

of Bachelor of Technology

In

Computer science and Engineering

Under the esteemed guidance of

Dr. M S MEKALA

By

Student-id	Name. Of Student
170031168	S. NITHISH KUMAR
170031297	T.SAI KARTHIK REDDY

(DST-FIST Sponsored Department)

K L EDUCATION FOUNDATION

Green Fields, Vaddeswaram, Guntur District-522 502

2019-2020

K L EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE ANDENGINEERING

(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that this project based lab report entitled **"Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure"** is a bonafide work done by "**17001297-T.Sai Karthik Reddy ,170031168 -S. Nithish Kumar**" in the course **Term Paper**, **17IE3247** in partial fulfilment of the requirements for the award of Degree in Bachelor of Technology in **COMPUTER SCIENCE AND ENGNEERING** during the Even Semester of Academic year 2019-2020.

Faculty in Charge **Head of the Department**

Dr. M S MEKALA **Mr. V. HARIKIRAN**

K L EDUCATION FOUNDATION

DEPT OF COMPUTER SCINCE AND ENGINEERING

(DST-FIST Sponsored Department)

s

DECLARATION

We here by declare that this project based lab report entitled **"Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure"** has been prepared by us in the course **Term Paper**, **17IE3247** in partial fulfilment of the requirement for the award of degree bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING** during the Even Semester of the academic year 2019-2020.We also declare that this project-based lab report is of our own effort and it has not been submitted to any other university for the award of any degree.

Date: 30-04-2020

Place: Vijayawada

Signature of the student:

170031168- S.NITHISH KUMAR

170031297- T.SAI KARTHIK REDDY

ACKNOWLEDGEMENT

Our sincere thanks to **Dr. M S MEKALA** in the Lab for her outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Dr. M S MEKALA** the instructor for the course **Term Paper, 17IE3247** in the Computer Science and Engineering Department for providing us with adequate planning and support and means by which we can complete this project-based Lab.

We express our gratitude to **Mr. V. HARIKIRAN,** Head of the Department for computer science and Engineering for providing us with adequate facilities, ways and means by which we can complete this project-based Lab.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the project-based Lab.

Last but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially our classmates and our friends for their support in the completion of our project-based Lab.

Name of the student

170031168- S. NITHISH KUMAR

170031297- T.SAI KARTHIK REDDY

TABLE OF CONTENTS

CHAPTERS

ABSTRACT

Facial expression recognition (FER) is an important type of visual information that can be used to understand a human's emotional situation. we propose a new scheme for FER system based on hierarchical deep learning. The feature extracted from the appearance feature-based network is fused with the geometric feature in a hierarchical structure. By this technique, we can extract the dynamic facial features between the neutral and emotional images without sequence data. This method combines the result of the softmax function of features by considering the error with prediction result. we propose a technique to generate facial images with neutral emotion using the auto encoder technique. By this technique, we can extract the dynamic facial features between the neutral and emotional images without sequence data. The ten-fold cross validation results show 96.46% of accuracy for CK+ dataset.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION:

Facial expression recognition (FER) is an important type of visual information that can be used to understand a human's emotional situation. In particular, the importance of AI systems has recently increased due to advancements in research on AI systems applied to AI robots.

Therefore, the most necessary technology is a vision sensor, as vision is a large part of human perception in most interactions.

A facial expression recognition system which accurately recognizes facial expressions in various environments has come to be more important and playing a vital role. Facial expression recognition systems (FERs) are fundamentally based on an evolutionary approach. Emotions in FER studies can be classified into six categories: happiness, sadness, fear, disgust, surprise, and anger.

1.2 PROBLEM DEFINITION:

In this Facial Expression Recognition system we are detecting the Facial expression using CNN model.The dataset taken is CK+ which is also called The Extended Cohn-Kanade Dataset consists of different types of emotion specified expressions. They are:

->Anger
->Contempt
->Disgust
→Fear
->Happy
->Sadness
->Surprise

1.3SCOPE:

Healthcare information systems tend to capture data in databases for research and analysis in order to assist in making medical decisions. As a result, medical information systems in hospitals and medical institutions become larger and larger and the process of extracting useful information becomes more difficult. Traditional manual data analysis has become inefficient and methods for efficient computer based analysis are essential. To this aim, many approaches to computerized data analysis have been considered and examined. Data mining represents a significant advance in the type of analytical tools currently available. It has been shown to be a valid, sensitive, and reliable method to discover patterns and relationships. It has been proven that the benefits of introducing data mining into medical analysis are to increase diagnostic accuracy, to reduce costs and to reduce human resources.

1.3 PURPOSE:

Facial expression recognition (FER) is an important type of visual information that can be used to understand a human’s emotional situation. In particular, the importance of AI systems has recently increased due to advancements in research on AI systems applied to AI robots.

Therefore, the most necessary technology is a vision sensor, as vision is a large part of human perception in most interactions.A facial expression recognition system which accurately recognizes facial expressions in various environments has come to be more important and playing a vital role. Facial expression recognition systems (FERs) are fundamentally based on an evolutionary approach. Emotions in FER studies can be classified into six categories: happiness, sadness, fear, disgust, surprise, and anger.

1.4 PROBLEM AND EXISTING TECHNOLOGY:

The idea of deep learning is a fast-growing and it works quite like a human mind. It represents the data in multiple levels and able to solve the selectivity-invariance dilemma efficiently . Deep learning techniques are used in a variety of forms in the field of medical prognosis. Many research works prove that deep learning techniques provide a better outcome, decline the classification error rate and more robust to noise than other strategies . It can handle the massive amount of data and have the capability to decode a complex problem in an easy way. Existing algorithms that extract facial landmarks, they detected the points of the eyebrows, eyes, nose, and mouth corners by applying the sobel edge, otsu algorithm, morphological operation, etc which is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background due to this the accuracy decreases and the pixel values may differ at some points due to thresholding process by this only we can get the skin colors. But in our system, the accuracy rate is much better than the state of the art, which is discussed in the result section.

1.6 PROPOSED SYSTEM:

Existing algorithms that extract facial landmarks, they detected the points of the eyebrows, eyes, nose, and mouth corners by applying the sobel edge, otsu algorithm, morphological operation, etc which is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background due to this the accuracy decreases and the pixel values may differ at some points due to thresholding process by this only we can get the skin colors. We compare the proposed algorithm with the other recent algorithms for CK+ dataset, which is typically considered to be verified dataset in the facial expression recognition. The ten-fold cross validation results show 96.46% of accuracy in the CK+ dataset.

CHAPTER 2: REQIUREMENTS & ANALYSIS

2.1 PLATFORM REQUIREMENTS:

Software Requirements :

The major software requirements of the project are as follows:
Language: python
Operating system: Windows
App: Anaconda Navigator, Jupyter Note Book.

Hardware Requirements:

The hardware requirements that map towards the software are as follows:
RAM: 4GB
Processor: Intel Core i3

2.2 MODULE DESCRIPTION:

1. Extraction of dataset .
2. Conversion of file format to list.
3. Resizing the image.
4. Converting to Gray images and Labelling the ranges.
5. Splitting the list into training and testing set.
6. Creation of CNN Model using Conv2D.
7. Giving the test set for model performance.
8. Storing the each epoch in a csv file for reference.
9. Get accuracy
10. print accuracy
11. Generating outputs classification.

CHAPTER 3: DESIGN & IMPLEMENTATION

3.1 ALGORITHMS:

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

3.2 PSEUDOCODE:

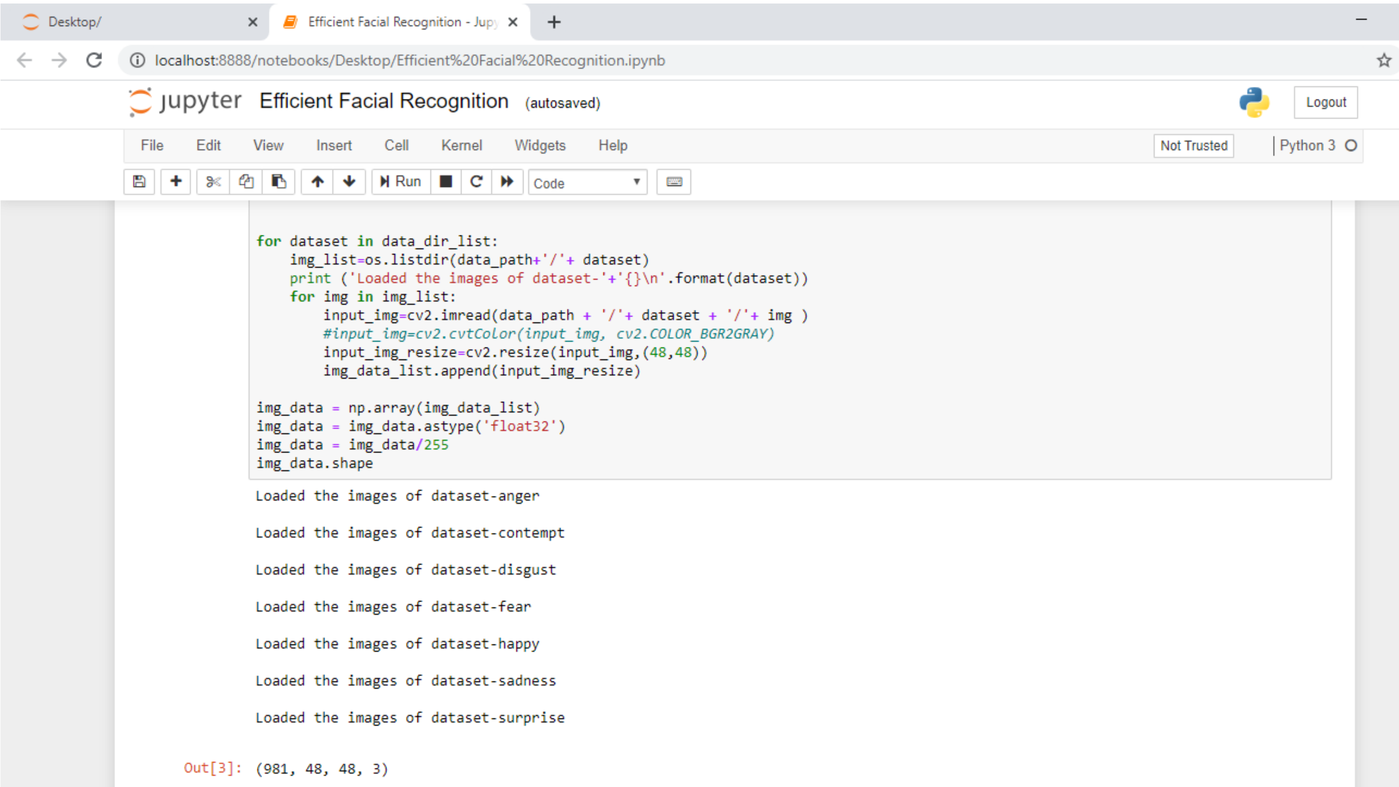
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os,cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from pylab import rcParams
rcParams['figure.figsize'] = 20, 10
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import keras
from keras.utils import np_utils
# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in
the input directory
```

```
from keras.models import Sequential
from keras.layers import Dense , Activation , Dropout ,Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.metrics import categorical_accuracy
from keras.models import model_from_json
from keras.callbacks import ModelCheckpoint
from keras.optimizers import *
from keras.layers.normalization import BatchNormalization
data_path = './ck/CK+48'
data_dir_list = os.listdir(data_path)
img_rows=256
img_cols=256
num_channel=1
num_epoch=10
img_data_list=[]
for dataset in data_dir_list:
img_list=os.listdir(data_path+'/'+ dataset)
print ('Loaded the images of dataset-'+'{}\n'.format(dataset))
for img in img_list:
input_img=cv2.imread(data_path + '/'+ dataset + '/'+ img )
#input_img=cv2.cvtColor(input_img, cv2.COLOR_BGR2GRAY)
input_img_resize=cv2.resize(input_img,(48,48))
img_data_list.append(input_img_resize)
img_data = np.array(img_data_list)
img_data = img_data.astype('float32')
img_data = img_data/255
img_data.shape
num_classes = 7
num_of_samples = img_data.shape[0]
labels = np.ones((num_of_samples,),dtype='int64')
labels[0:134]=0 #135
labels[135:188]=1 #54
labels[189:365]=2 #177
labels[366:440]=3 #75
```

```
labels[441:647]=4 #207
labels[648:731]=5 #84
labels[732:980]=6 #249
names = ['anger','contempt','disgust','fear','happy','sadness','surprise']
def getLabel(id):
return ['anger','contempt','disgust','fear','happy','sadness','surprise'][id]
Y = np_utils.to_categorical(labels, num_classes)
#Shuffle the dataset
x,y = shuffle(img_data,Y, random_state=2)
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.15, random_state=2)
x_test=X_test
input_shape=(48,48,3)
model = Sequential()
model.add(Conv2D(6, (5, 5), input_shape=input_shape, padding='same', activation =
'relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(16, (5, 5), padding='same', activation = 'relu'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation = 'softmax'))
model.compile(loss='categorical_crossentropy', metrics=['accuracy'],optimizer='adam')
model.summary()
model.get_config()
model.layers[0].get_config()
model.layers[0].input_shape
model.layers[0].output_shape
model.layers[0].get_weights()
np.shape(model.layers[0].get_weights()[0])
```

```
model.layers[0].trainable
from keras import callbacks
filename='model_train_new.csv'
filepath="Best-weights-my_model-{epoch:03d}-{loss:.4f}-{acc:.4f}.hdf5"
csv_log=callbacks.CSVLogger(filename, separator=',', append=False)
checkpoint = callbacks.ModelCheckpoint(filepath, monitor='val_loss', verbose=1,
save_best_only=True, mode='min')
callbacks_list = [csv_log,checkpoint]
callbacks_list = [csv_log]
score = model.evaluate(X_test, y_test, verbose=0)
print('Test Loss:', score[0]*100)
print('Test accuracy:', score[1]*100)
test_image = X_test[0:1]
print (test_image.shape)
print(model.predict(test_image))
print(model.predict_classes(test_image))
print(y_test[0:1])
res = model.predict_classes(X_test[9:18])
plt.figure(figsize=(10, 10))
for i in range(0, 9):
plt.subplot(330 + 1 + i)
plt.imshow(x_test[i],cmap=plt.get_cmap('gray'))
plt.gca().get_xaxis().set_ticks([])
plt.gca().get_yaxis().set_ticks([])
plt.ylabel('prediction = %s' % getLabel(res[i]), fontsize=14)
# show the plot
plt.show()
```

CHAPTER 4: SCREENSHOTS

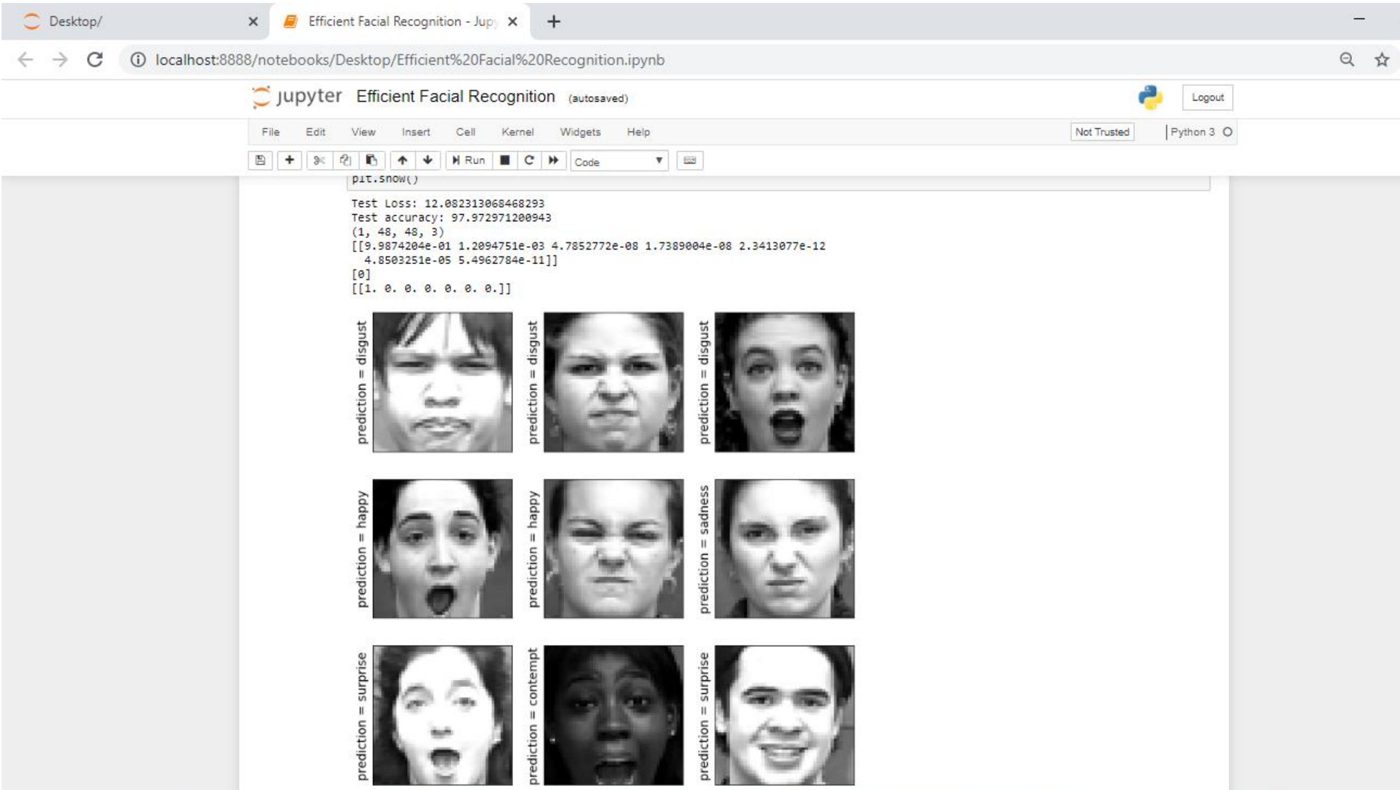


```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 6)         456
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 24, 24, 6)         0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 16)        2416
_________________________________________________________________
activation_1 (Activation)    (None, 24, 24, 16)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 10, 10, 64)        9280
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               204928
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 903
=================================================================
Total params: 217,983
Trainable params: 217,983
Non-trainable params: 0
_________________________________________________________________

Out[7]: True
```

```python
filename='model_train_new.csv'
filepath="Best-weights-my_model-{epoch:03d}-{loss:.4f}-{acc:.4f}.hdf5"

csv_log=callbacks.CSVLogger(filename, separator=',', append=False)
checkpoint = callbacks.ModelCheckpoint(filepath, monitor='val_loss', verbose=1, save_best_only=True, mode='min')
callbacks_list = [csv_log,checkpoint]
callbacks_list = [csv_log]
```

```
In [9]: hist = model.fit(X_train, y_train, batch_size=7, epochs=50, verbose=1, validation_data=(X_test, y_test),callbacks=callbacks_list
```

```
Train on 833 samples, validate on 148 samples
Epoch 1/50
833/833 [==============================] - 4s 5ms/step - loss: 1.7924 - accuracy: 0.3097 - val_loss: 1.5485 - val_accuracy:
0.5743
Epoch 2/50
833/833 [==============================] - 2s 3ms/step - loss: 1.1850 - accuracy: 0.5654 - val_loss: 0.8467 - val_accuracy:
0.7568
Epoch 3/50
833/833 [==============================] - 2s 3ms/step - loss: 0.8043 - accuracy: 0.6939 - val_loss: 0.6550 - val_accuracy:
0.8311
Epoch 4/50
833/833 [==============================] - 3s 4ms/step - loss: 0.6437 - accuracy: 0.7671 - val_loss: 0.4377 - val_accuracy:
0.8514
Epoch 5/50
833/833 [==============================] - 2s 3ms/step - loss: 0.4787 - accuracy: 0.8307 - val_loss: 0.3266 - val_accuracy:
0.9054
Epoch 6/50
833/833 [==============================] - 2s 3ms/step - loss: 0.3443 - accuracy: 0.8956 - val_loss: 0.2377 - val_accuracy:
```



CHAPTER 5: CONCLUSION

We have proposed an efficient facial expression recognition algorithm combining appearance feature and geometric feature based on deep neural networks for more accurate and efficient facial expression recognition. The appearance feature-based network extracts the holistic feature of the LBP feature containing the AUs information. The geometric feature-based network extracts the dynamic feature, which is the face landmark change centered on the coordinate movement between the neutral face and the peak emotion. As a result, we constructed more robust feature by combining static appearance feature from the appearance network and dynamic feature from the geometric feature-based network. In the experiments, we have shown that the Top-2 error frequently occurred with average about 82% using only appearance feature-based network. As a result of improving this error with the proposed algorithm, we achieved about 96.5% accuracy with 1.3% improvement when comparing to the other algorithms in the CK+ dataset.

CHAPTER 6: REFERENCES

1) **->"Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure" -** https://ieeexplore.ieee.org/document/8673885/metrics#metrics

2) **-> ''Facial expression recognition using hierarchical features with deep comprehensive multipatches aggregation convolutional neural networks**,''**-** https://ieeexplore.ieee.org/document/8673885/metrics#metrics