**Name: Pravin Reddy Sereddy**

**Email: spravin1@binghmaton.edu**

**Purpose of the Project:**

The main purpose of the project is getting the knowledge of pattern recognition in the images using the k means and KNN algorithm and motion detection of an object in two sequential frames.

And all these methods, implementations and results are explained in the below section

**Programs required environment setup and Running the Programs:**

There are 2 program files in this zipped file

* **p1.cpp**
* **p2.cpp**

**Required Environment setup:**

* Opencv 3.4
* Linux based system with g++ compiler for compiling the c++ programs

**Running the program:**

Select the above program required and compile the program in the terminal with above environment

**For compiling:**

g++ -ggdb filename.cpp -o filename ‘pkg-config –cflags –libs opencv’

**For running:**

./filename image

**Programming Design**

**Methods used:**

**KNN Algorithm:**

* I have loaded the image into my program
* From the image I have sub divided the image into 4x4 blocks and stored into a matrix
* Now I have applied the KNN algorithm
* And I have initialized the k values to the 0, 128, 255
* For training the data I have manually taken the average of each sub images 4x4 and compared to the nearest class and have assigned the class to that block and each pixel in that block is filled with the new class value
* For predicting the classes I have taken the averages of each sub image pixels then I have calculated the Euclidean distance between the test block from the bottom half and to each block from the top half of the image.
* Get all the distances from the above process and sort them in the ascending order
* Now get the top 50 nearest distances and count the frequencies of each class from the nearest 50 distances and take class that has highest frequencies and this class is the predicted class for the image

**K-means Algorithm:**

* I have loaded the image into my program
* From the image I have taken each pixel intensity value and stored into a matrix
* Now I have applied the K-means algorithm
* And I have considered the classes as 0, 128, 255
* Now I have calculated the distance between the class value to each pixel
* Now take all the distances and now cluster the pixels to the nearest pixel values of the class values
* Now replace each cluster value with the class value and output the image

**Motion detection Algorithm:**

* I have loaded the image into my program
* From the image I have converted the colour image to the greyscale image
* Now I have sub divide the image to the 8x8 blocks and classified the block into three classes 0, 128, 255 using the KNN algorithm
* From the two images I have searched for the reference blocks for the best matched block in the frame to another frame which we call as the reference blocking
* Now take the center of the block from the first frame and draw a vector to the block to where it has moved to in the second frame
* This results the motion vectors in the image of the object movement and this is the output.

**Results:**

**M1**

**A picture containing photo

Description automatically generated**

**M2**

**A picture containing screenshot

Description automatically generated**

**N2**

**A screenshot of a cell phone

Description automatically generated**

**N3**

**A screenshot of a computer

Description automatically generated**

**N4**

**A close up of a logo

Description automatically generated**

**T1**

**A picture containing screenshot

Description automatically generated**

**Error rate:** A close up of a sign

Description automatically generated

**K1**

**A picture containing photo

Description automatically generated**

**Block image of 1st frame for motion detection**

**A picture containing screenshot

Description automatically generated**

**Block image of 2nd frame for motion deteection**

**A picture containing screenshot

Description automatically generated**

**5. Bug Report**

* No, bugs we detected to my knowledge