caMicroscope

Proposal - Google Summer of Code 2020

Project Proposal

1. Project Title

Cancer Region of Interest Extraction and Machine Learning

2. Abstract

From what I have gathered from my research over the internet, caMicroscope is an open-source digital pathology data management, visualization and analysis platform. As of now, It supports single-stage networks showing results directly. The objective is to shift to

It basically is a tool to view, annotate, and analyze biomedical images.

After going through the public website of caMicroscope, I got the gist of the work that they do. Generally for such projects, Histopathological images are used from datasets like The Breast Cancer Histopathological Image Classification (BreakHis) and classes are created like “Benign” and “Malignant” for the detection of cancer.

What this project seeks is that once a region of interest is extracted from the image it should be downloaded and fed to other models for tasks like synthetic data generation. The aim is to support a wider range of models and flexible use of their outputs.

3. Student Details

- Name: Hrishabh Digaari
- Slack Username: Hrishabh Digaari
- Email ID: hrishbabhdigaari@gmail.com
- Telephone : +91 8800500335
- Country of Residence: India
- Timezone: IST (India)
- Language: English

4. Potential Mentors

Insiyah Hajoori and Ryan Birmingham

5. Personal Background

I am a third-year Bachelors of Technology Undergraduate student of Computer Science Engineering with an interest in Neural Networks.

Projects / Freelancing / Internships

Duration (From – till)	**Organization**	**Title**	**Concepts/Technology used**
January 2020 – Present	Manipal University Jaipur	Occlusion detection and gait silhouette reconstruction from degraded scenes	Deep Learning Pipeline, Convolutional Neural Networks(Alexnet, VGG 16), Yolov3, Python, Tensorflow, Tensorflow.js, Flask, Transfer Learning
October 2019 - Present	Barrow (My own Startup)	Food Ordering and Expense Tracker App	Flutter, Python
July 2019– Dec 2019	Freelance	Course and Assignments sharing platform for the faculty of universities	Used Reactjs for Frontend
May 2019 – Jul 2019	IITD-AIA Foundation for Smart Manufacturing (IAFSM)	Industrial IoT 4.0 Research and Development Internship	• Designed, Developed and Deployed an IoT Dashboard for monitoring industrial systems. • ReactJS was used to develop the dashboard. • The service was linked with real-time Firebase

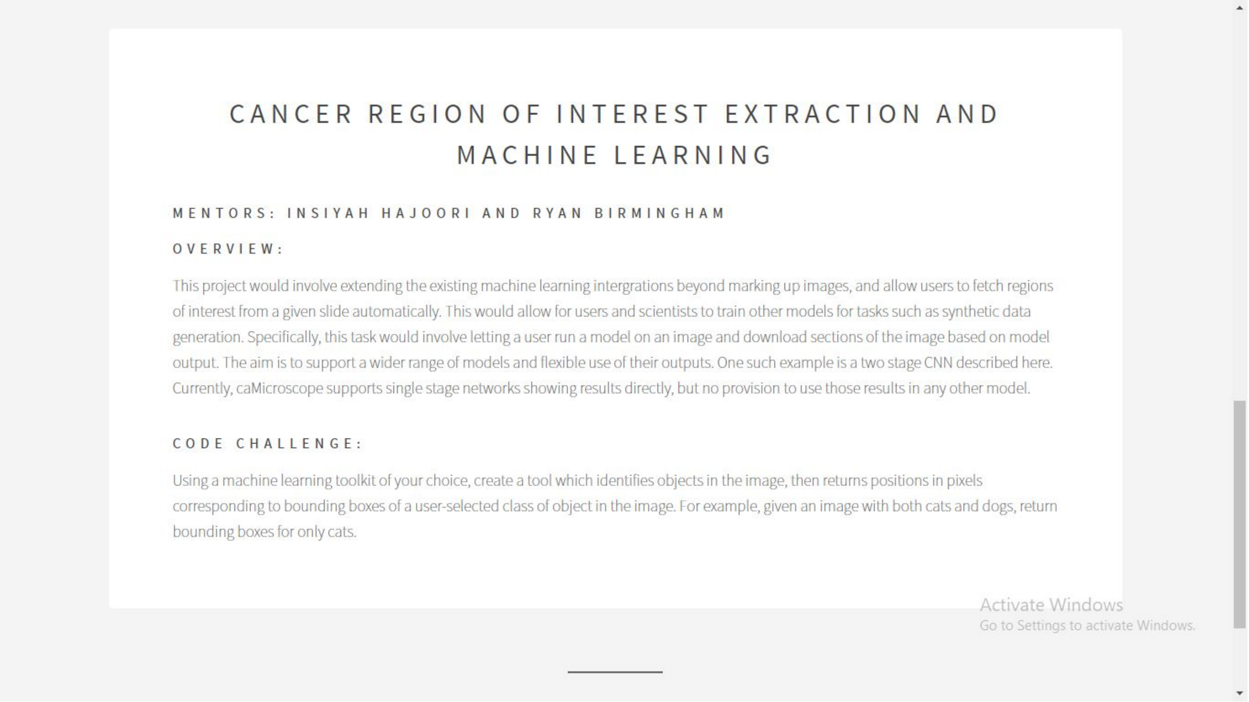
			database • Hands-on experience in OPC Unified Architecture and Node-RED

6. Code Challenge/Work done till now

We were given the challenge to make a project on interest extraction Using a machine learning toolkit of our choice. The project was to create a tool which identifies objects in the image, then returns positions in pixels corresponding to bounding boxes of the object in the image.

Here is the GitHub Repo: https://github.com/Digaari/caMicroscope

And following are the Snapshots:



CANCER REGION OF INTEREST EXTRACTION AND MACHINE LEARNING

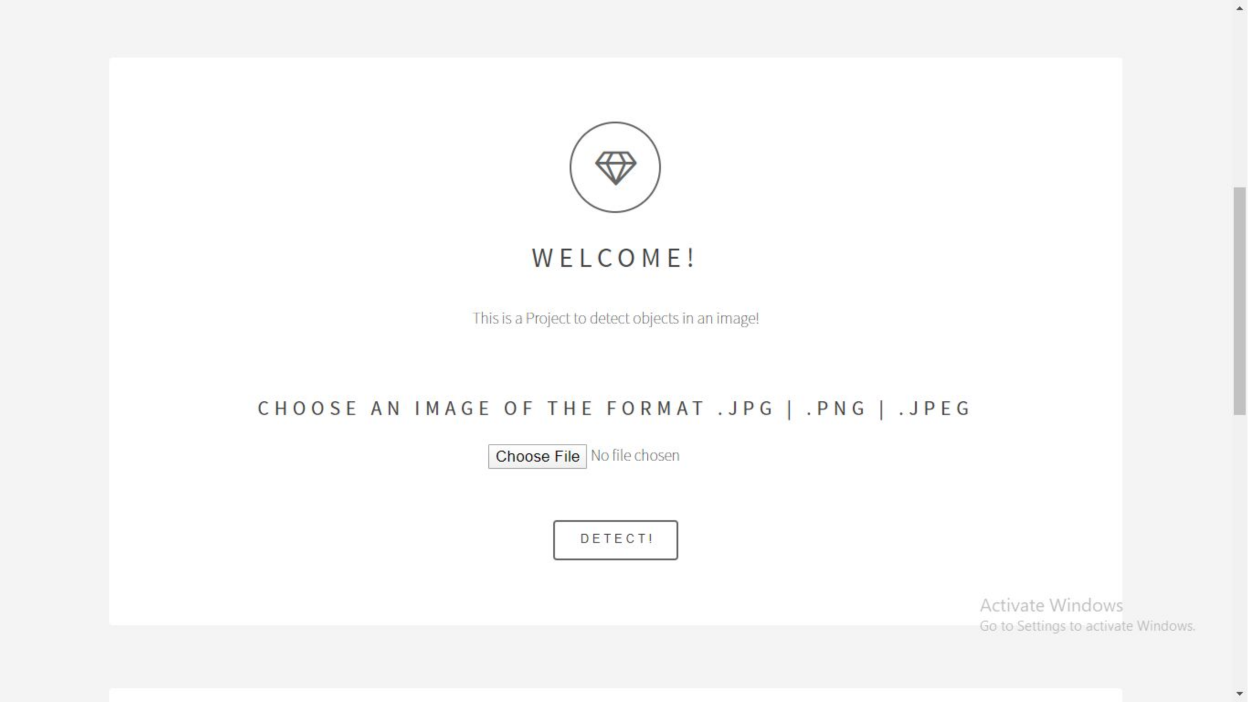
MENTORS: INSIYAH HAJOORI AND RYAN BIRMINGHAM

OVERVIEW:

This project would involve extending the existing machine learning intergrations beyond marking up images, and allow users to fetch regions of interest from a given slide automatically. This would allow for users and scientists to train other models for tasks such as synthetic data generation. Specifically, this task would involve letting a user run a model on an image and download sections of the image based on model output. The aim is to support a wider range of models and flexible use of their outputs. One such example is a two stage CNN described here. Currently, caMicroscope supports single stage networks showing results directly, but no provision to use those results in any other model.

CODE CHALLENGE:

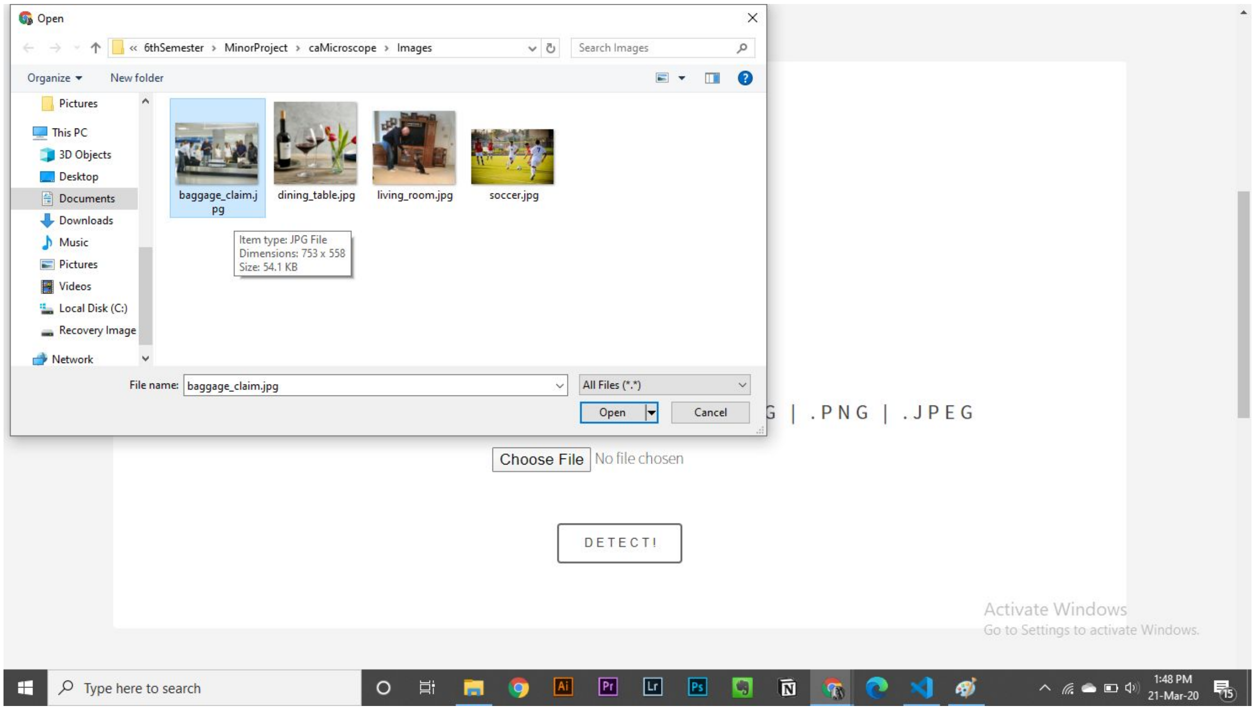
Using a machine learning toolkit of your choice, create a tool which identifies objects in the image, then returns positions in pixels corresponding to bounding boxes of a user-selected class of object in the image. For example, given an image with both cats and dogs, return bounding boxes for only cats.



WELCOME!

This is a Project to detect objects in an image!

CHOOSE AN IMAGE OF THE FORMAT .JPG | .PNG | .JPEG

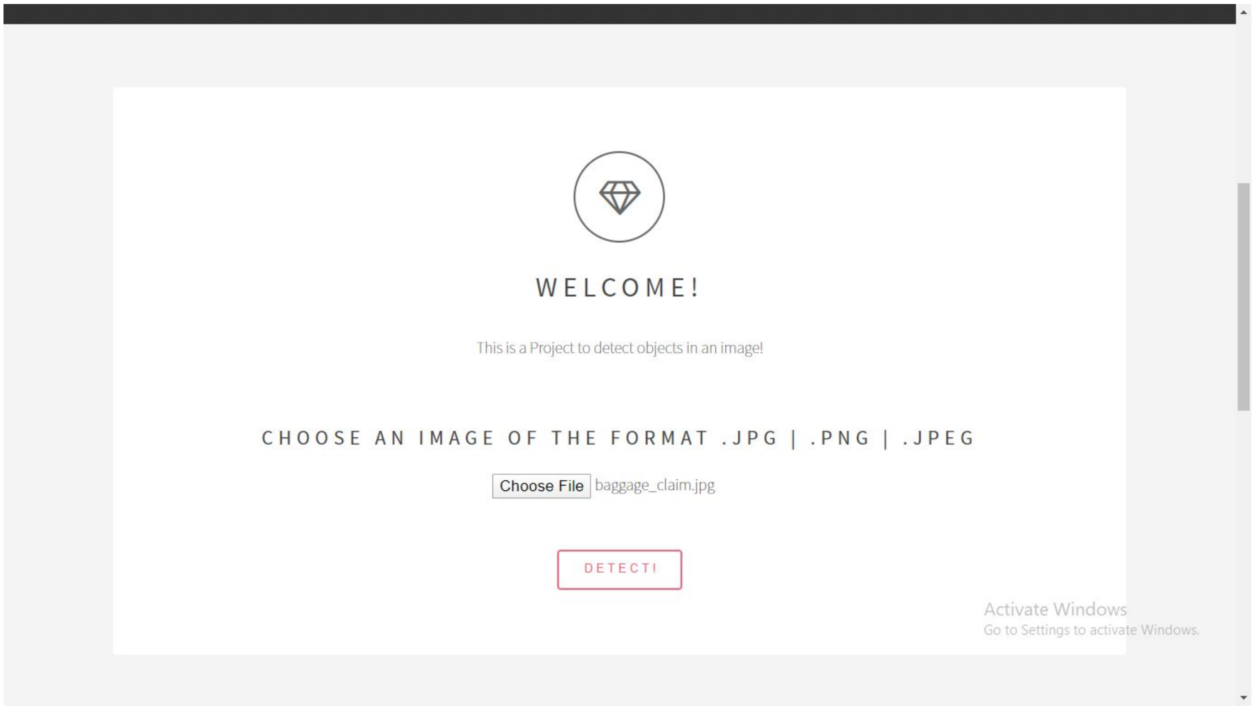


Item type: JPG File
Dimensions: 753 x 558
Size: 54.1 KB

File name: baggage_claim.jpg

All Files (*.*)

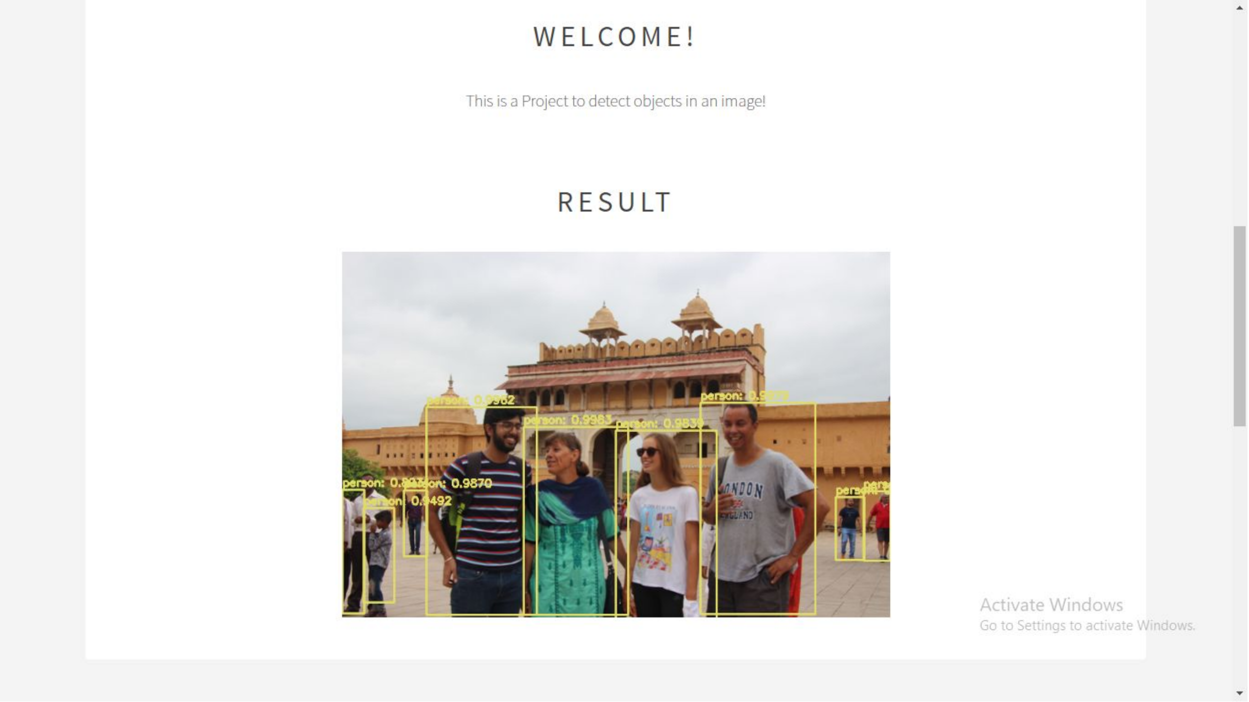
No file chosen


WELCOME!

This is a Project to detect objects in an image!

CHOOSE AN IMAGE OF THE FORMAT .JPG | .PNG | .JPEG

Choose File baggage_claim.jpg



WELCOME!

This is a Project to detect objects in an image!

RESULT

I started on this project on the 23rd of February. Here are my error/progress logs:

LOGS START

23rd February 2020

Started my research on various other CNN architectures and object detection models.

29th February 2020

Created a workflow for me. Divided my project into:

1. Architecture Design
    a. Create the layers of the network architecture

   b. Implement the forward pass of the network
   c. Objectness Confidence Thresholding and Non-maximum Suppression
   d. Design the input and the output pipelines
2. Data collection
   a. kaggle
3. Data preparation (even though I’ll use Kaggle)
   a. Verify the annotations are correct
   b. Ensure the EXIF orientation of the images is correct
   c. Resize images and update image annotations to match the newly sized images
   d. Colour corrections
   e. Format annotations to match the requirements of the model’s inputs
4. Model training and improvement
   a. Will use Google colab
   b. Load in the data via the Kaggle
   c. Determine the model configuration
   d. Initiate training
   e. Use the trained model for inference
5. Deploy using Flask and firebase
6. Give a front end
7. Detect!

2nd March 2020

Used the official cfg file from the author for the network.

USed OpenCv. Parsing through cfg.

5 types of layers: Convolutional, Shortcut, Upsample, Route, YOLO.

Implemented the forward pass of the network

Testing the forward pass: failed

Can't store the image.

4th March 2020

Testing forward pass:

```
   blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
                                 swapRB=True, crop=False)
    net.setInput(blob)
    start = time.time()
    layerOutputs = net.forward(ln)
    print(layerOutputs)
    end = time.time()
```

```
(  0  ,.,.) =
   16.0962   17.0541   91.5104  ...     0.4336    0.4692    0.5279
   15.1363   15.2568  166.0840  ...     0.5561    0.5414    0.5318
   14.4763   18.5405  409.4371  ...     0.5908    0.5353    0.4979
                ⋱                 ...
  411.2625  412.0660    9.0127  ...     0.5054    0.4662    0.5043
  412.1762  412.4936   16.0449  ...     0.4815    0.4979    0.4582
  412.1629  411.4338   34.9027  ...     0.4306    0.5462    0.4138
[torch.FloatTensor of size 1x10647x85]
```

Downloaded weights and loaded them

```
labelsPath="gsoc_model/coco.names"

cfgpath="gsoc_model/gsoc.cfg"
```

```
wpath="gsoc_model/gsoc.weights"

Lables=get_labels(labelsPath)

CFG=get_config(cfgpath)

Weights=get_weights(wpath)

nets=load_model(CFG,Weights)

Colors=get_colors(Lables)
```

As of now: built the model and weights loaded

Can finally start detecting objects.

10th March 2020
Finally started with detection. Wrote `app.py`

12th march 2020
Debugged app.py
Deployed it using flask

15th March 2020
Finally got the image on the server. Used Pyrebase4
Do not use Pyrebase

LOGS END

---

7. Project Schedule

1. 28th April - 18th May: **Community Bonding**
    a. Community Bonding.
    b. Understanding the structure of caMicroscope.
    c. Getting familiar with all the relevant technologies/ concepts.
2. 19th May - 10th June: **Development Phase**
    a. Development starts
    b. Reviewing the whole architecture
    c. Making the necessary additions to the pre-existing model
3. 15th June - 19th June: **Phase 1 Submission**
    a. submitting Phase 1 evaluations
    b. Inline comments
    c. Comment on a selection
4. 20th June - 13th July - **Development Phase**
    a. Development resumes
    b. Reviewing the whole architecture
    c. Making the necessary additions to the pre-existing model
5. 13th July - 17th July - **Phase 2 Submission**
    a. submitting Phase 2 evaluations
    b. Inline comments
    c. Comment on a selection
6. 18th July - 31st July - **Allow admin to limit access to specific files**
    a. Up to this point, a user will have common permission (View/Edit/Comment) to access all files.
    b. Add feature to allow users to provide a different level of access to different files.
7. 1st Aug - 9th Aug -**Project Completion, testing, and documentation**
    a. Wrapping up the project
    b. Documenting
8. 10th Aug - 17th Aug - **Final Testing**
    a. Final evaluation

8. Planned absence/vacation days and other commitments during the GSoC period

I have my summer vacation till 31st May 2020 due to COVID-19. My college will be reopening on 1st June 2020. My college timings are 9am to 5pm (IST). I have planned a working schedule for myself. Working hours are shown below.

9. Planned GSoC work hours

My prefered working hours are:

Indian Standard Time (IST)	Coordinated Universal Time(UTC)	Total Time
12:00pm-13:30pm	6:30am-8:00am	1.5 hours
18:00pm-22:00pm	12:30pm-16:30pm	4 hours
23:00pm-03:00am	17:30pm-21:30pm	5 hours

10. Skill Set

I have done Courses on Neural Networks and Convolutional Neural Networks from Coursera.

Have worked on projects with Tensorflow and Tensorflow.js

Currently, I am working on GAIT analysis: Occlusion detection and gait silhouette reconstruction from degraded scenes using deep learning wherein I have prepared a customized layer architecture for our dataset provided by the Technical University of Munich and IIT Kharagpur (TUM-IITKPG) following is the gist of the work:

Stage 1: Background removal

In image processing there usually is preprocessing of the image where the first step is segmentation. This refers to where the background is separated from the foreground to isolate the foreground.

The background removal techniques:

1. Non-recursive technique

2. Recursive technique

Stage 2: Feature extraction

Once the background is subtracted from the image, each image sequence is converted into a temporary sequence of distance signals. A feature of an image is represented using a vector.

Two approaches for feature extraction

Approach 1: Model-based approach

- high-quality video sequences
- parameters used: height, the distance between the pelvis and feet and the distance between the feet.
- The silhouette is divided into some regions that are feature vectors that include averages of the centroid, the aspect ratio and orientation of the major axis of the ellipse

Approach 2: Holistic approach

- More focus on the shape of the silhouette or the motion of the whole body as compared to model-based approaches that focus on a particular part of the body.
- not dependent on the quality of the video frame.
- offer less computational requirements and complexities
- Not as robust as compared to model-based approaches
- contour of the silhouette is regarded as an important factor of the method.
- For high-quality silhouette, the outer contour of the silhouette is regarded as the important feature

- For low-quality silhouette, the binaries silhouettes are regarded as the important feature

Methodology

- ➢ Scrutinizing the data
- ➢ Automated identification of humans from their gait is a challenging research problem.
- ➢ We consider situations where the subject gets occluded due to the presence of multiple objects in the field of view of the camera, which is quite common in real-world surveillance scenarios.
- ➢ There are two types of occlusions: Static and Dynamic.
- ➢ Upon research, we came across a novel approach for detecting the presence of occlusion in a sequence of silhouette frames and their subsequent reconstruction.
- ➢ Dynamic programming- based maximum likelihood key pose detection algorithm simultaneously detects key pose class for each frame and also identifies the occluded frames.
- ➢ Clean and unclean gait cycles are segregated as the algorithm output. If all the subsequences of frames corresponding to a gait cycle are degraded by occlusion, then none of the existing methods can be used for recognition from this sequence.
- ➢ Need for the reconstruction of the degraded silhouette frames to construct clean gait cycles becomes pertinent in such situations.

- ➢ A novel method based on BGPDM which is able to reconstruct the missing silhouettes considerably well.
- ➢ The reconstructed silhouettes can then be used for recognition using any of the existing methods. We will test our algorithms on the data set (TUM-IITKGP) featuring occlusion by static objects as well as dynamic objects.