# Report 5

# Embedded Machine Vision

# Intelligent Automation

# Extended Lab

By: Omkar Prabhu

Partner: Nikhil Divekar

**Q1.** Article talks about DARPA grand challenge 2005 about building autonomous robot and train it to drive on terrain which is new. Using different machine learning techniques participants must train the robot.

There were so many varying conditions due to dirt road and rocky mountain pass and participants are not allowed to train robot on path until 2 hours for the competition. Faster robot wins the race.

Stanford racing team developed stanley robot. They used different sensors like 2 Radar sensors, GPS module to train robot to act according to topographical conditions and air resistance. They were using variable air suspension, four-wheel drive, variable-, and electronic locking according to sensors input.

The system was divided into stack of 4 layers

- Sensing
- Analysis of the data
- Control outputs(actuators) accordingly
- Vehicle User Interface

Sensing:

Color Cams and Lasers were used for sensing (obstacle and surface detection and identifcation).

**Q2.** Since we are submitting 3 different proposals, I have worked on pedestrian detection as well as creating VARI based false NDVI image for vegetation health analysis.

**Pedestrian Detection:**

**Approach:**

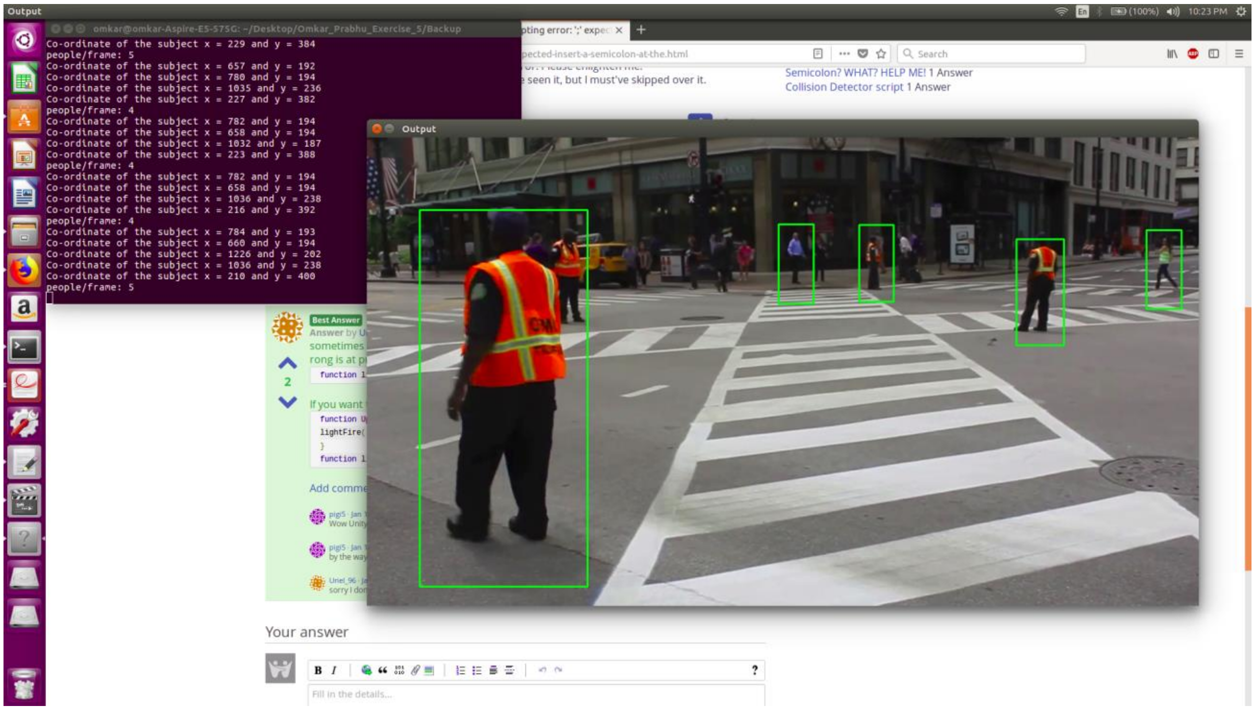
Used Histogram of Gradients used for people detection, The image was first segmented 1:2 ratio (typically 64:128) and resized.

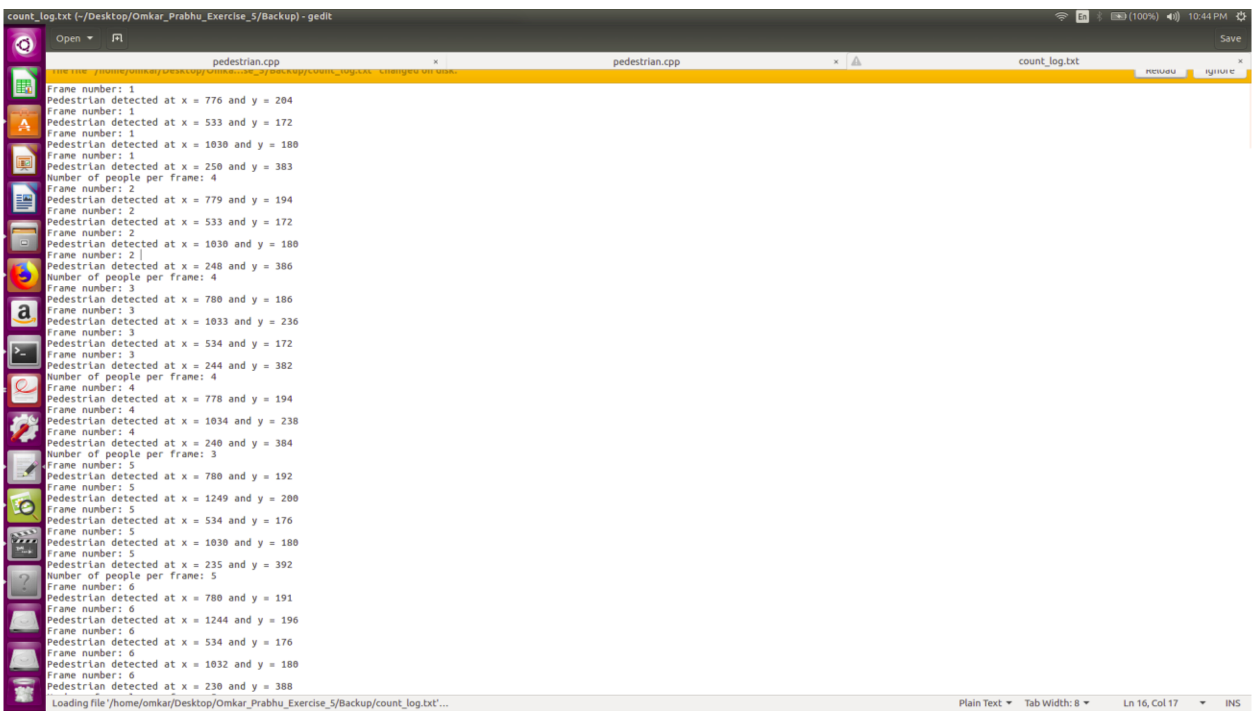
Then image is divided into smalled 8*8 blocks to calculate gradients using sobel over that

Once the sobel gradients are calculated they are re-created to form histogram using angles as bins and adding total magnitude on Y-axis according to the angle to get the resultant gradient mag and direction

The resultant gradient would be alligned according to persons shape which can be used for tracking

## Logging and Tracking




NDVI Index Calulation:

**Approach:**

NDVI index is hard to calculate if NIR spectrum is not present in the band since the formula is

NDVI = (NIR-RED)/(NIR+RED)

So we need to get information about 4th channel I.e NIR

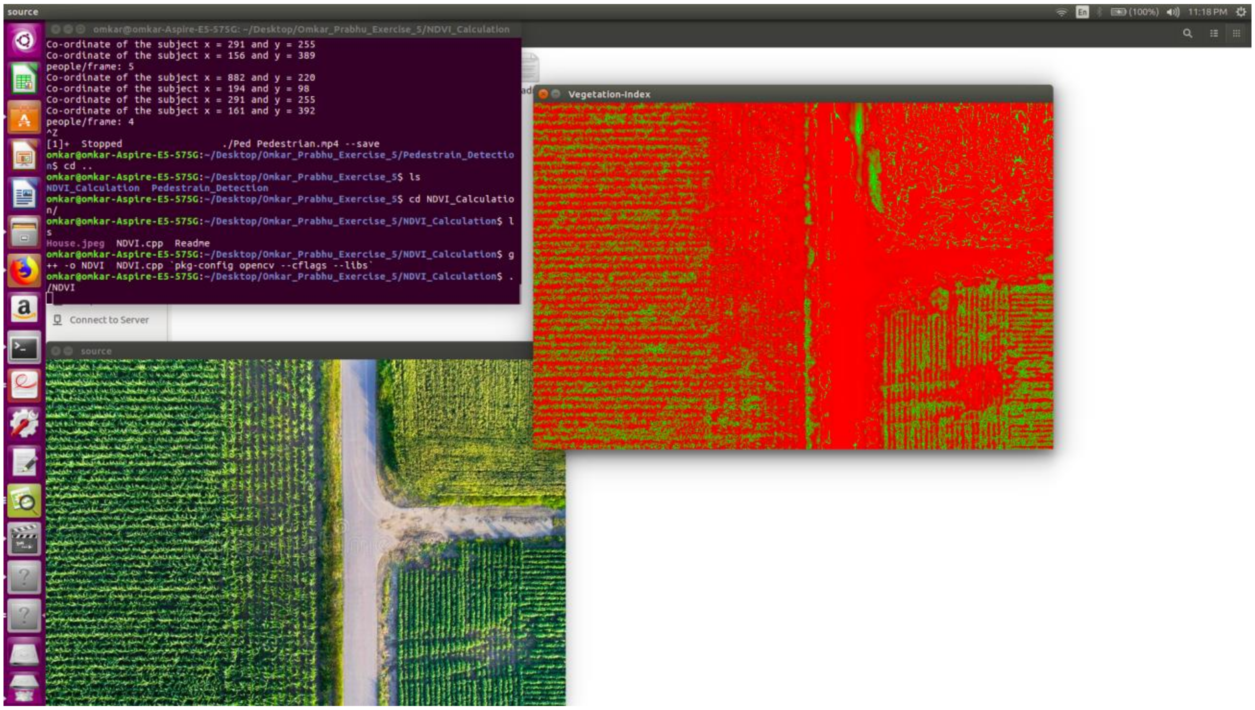
Hence to calculate it without NIR we use false NDVI which is VARI which can be used on RGB cameras mounted on drones

Used pixel manipulation and the formula given below to calculate VARI

VAI = (GREEN-RED)/(GREEN+RED-BLUE)

**RGB Image of a farm:**

VARI (Green Areas represents healthy plants while red and yellow areas represent water and land)




Q3.

We are currently deciding among 3 ideas so far this exercise, we are submitting 3 different proposals, titled Option 1, Option 2, Option 3.

**Option 1:**

Self-driving car:

Idea and Implementation:

- Self-driving car is the vehicle with an autopilot system and capable of driving from one point to another without human intervention.
- We are planning to implement the self-driving car system from the image processing point of view. The major modules which would be included in the project are lane detection, vehicle and pedestrian detection along with road sign detections.
- We plan to control the steering module and perform the speed control according to the lanes detected and other vehicles and road signs.
- Lane detection will be helpful in deciding the movement of steering wheel in right or left to keep the vehicle in the same lane.
- Vehicles in the front are detected and speed will be reduced if there is vehicle stopped right in the front or if the vehicle ahead of us is decelerating. Vehicle will also stop if stop sign or red traffic signal is detected.
- We also plan to integrate this module with Google maps APIs so that self-driving car should be able to go from one point to other following the directions provided by the google maps.
- Also, lane detection module is used to understand whether it is valid to overtake the vehicle right in front of us. Lane departure warning system can also be implemented for safety purposes.

- Rear camera is also integrated to confirm the safety while lane changing or turning onto the different road.

Relevance of selection:

- Many big company’s like Google and Tesla are prominent leaders in self-driving car. We plan to implement the self-driving car module prototype with different modules. These modules will collectively achieve target of automatic steering and speed control, reacting according to the road signs and finding route from source to destination.
- OpenCV can be used to detect the lane (hough transform), vehicles (haar cascade) and pedestrians (HOG descriptors), Also machine learning can be integrated to identify the road signs and also for affiliation with Google map APIs.

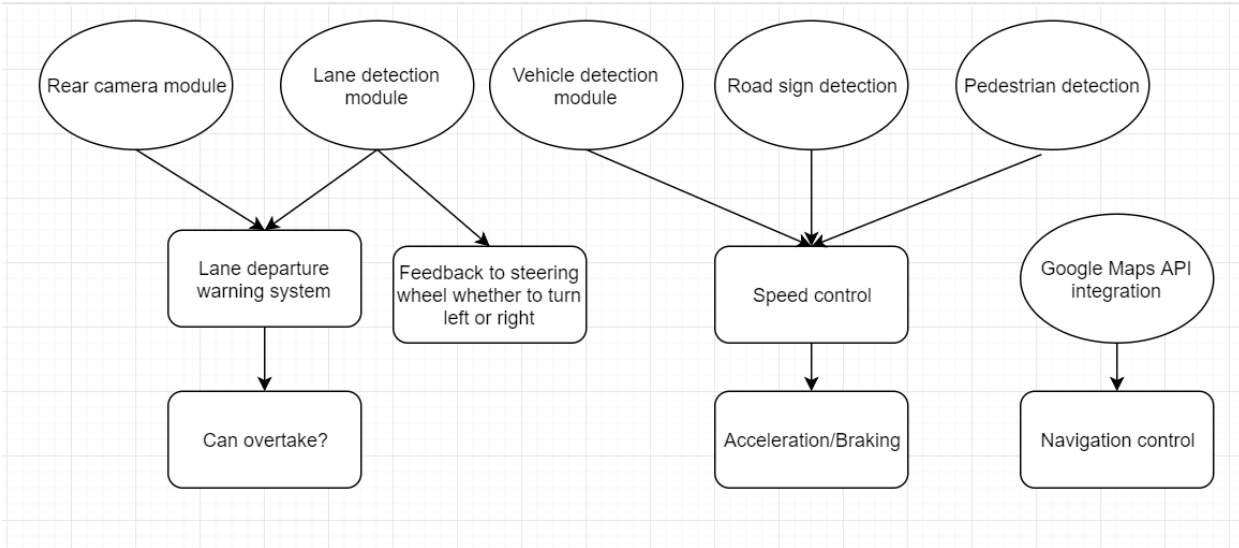
Challenge:

- The biggest challenge with this problem statement should be the integration of all the above-mentioned modules.
- The efficiency and speed should be kept optimal even after integrating the different modules. We are planning to use multi-threading to minimize the speed concern.
- Also, exact identification of road signs and integration with google map APIs provides challenge to some extent.

Rationale of selection:

1. Minimization of human intervention and thereby reducing the chances of human mistakes and thus fewer accidents.
2. Traffic management will be improved to a greater extent and thus decreased traffic congestion.
3. Lower fuel consumption due to minimization of traffic jams.
4. This will also minimize the incident of speeding since car will be programmed to follow the road signs and thus also maintain speed according to the limit.

Block diagram:




**Option 2**

<u>Vehicle speed detection and license plate extraction using OpenCV:</u>

Idea and Implementation:

- Speeding endangers everyone on the road: In 2016, speeding killed 10,111 people, accounting for more than a quarter (27%) of all traffic fatalities that year. Thus, the automatic system which can help keep check on speeding should be implemented.
- The basic idea of this project is the speed detection of the vehicles on the road and license plate extraction of the vehicle which is navigating above the speed limit.
- We are planning to use two cameras. One camera will be at certain height so that it should be able to capture the video of the traffic on the road. The main functionality of this camera is to detect the vehicles moving in one direction and track their speed.
- We will also capture the image of the vehicle and track the count of vehicles according to their size whether its light weight or heavy weight vehicle. Now, if certain vehicle is observed to go at the speed higher than the speed limit, then the second camera placed at certain distance from the first one and at the lower height will be triggered.
- According to the speed and distance between two cameras, second camera will properly capture the image of the speeding vehicle. Since this camera is at the lower altitude, it will be helpful in getting proper look at the number plate.
- Once the image is captured, we plan on marking the number plate of that vehicle as the region of interest and extract it from the image of the car. Further, image enhancement and sharpening will be used to detect the characters and numbers from the number plate.

- The ultimate target will be exact determination of the plate number of the vehicles and then logging the speeding vehicle's license plate number along with recorded speed and date and time to a certain log file.

Relevance of selection:

- The OpenCV can be utilized in vehicle speed tracking and pinpointing of speeding vehicles. Vehicle detection can be achieved by Haar cascade.
- Speed can be detected by following the line once vehicle is tracked. This implementation can be integrated with machine learning for exact determination of the number plates.

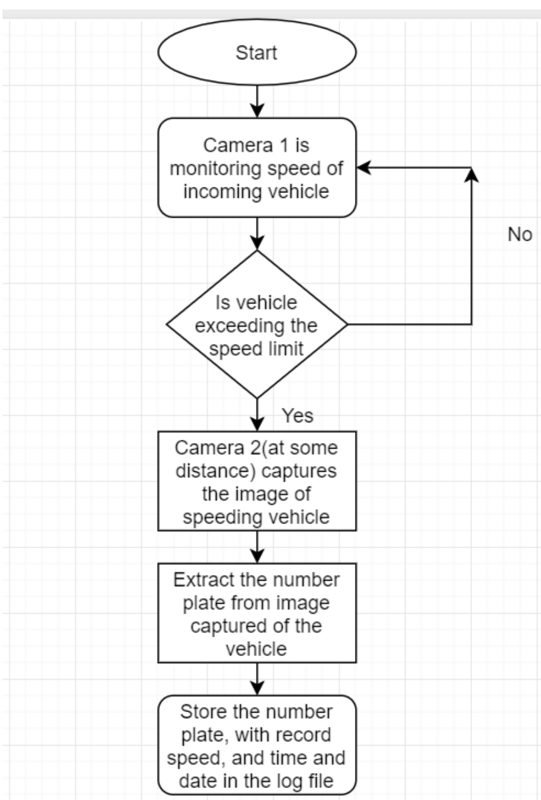
Challenge:

- The utmost challenge in the implementation is the synchronization of the two cameras. One which records the speed of all the vehicles and other one which captures the image of the speeding vehicle.
- Also, another challenge includes the exact determination of the number plate of the speeding car when zoomed-in.

Rationale of the selection:

5. This implementation helps to eliminate the efforts of police patrolling the major roads. It reduced the human labor and thereby reduces the probability of mistakes.
6. The implementation provides with very important details of speed tracking and catching the speeding vehicle which can be stored in database for future data analytics.
7. Further, the number plate can be tracked back to owner and speeding ticket can be directly issued online using certain database. (Future Scope).

Flow-chart:


Start
Camera 1 is monitoring speed of incoming vehicle
No
Is vehicle exceeding the speed limit
Yes
Camera 2(at some distance) captures the image of speeding vehicle
Extract the number plate from image captured of the vehicle
Store the number plate, with record speed, and time and date in the log file

**Option 3:**

Precision Agriculture using Image processing

- Precision agriculture is the method where you analyze certain large-scale farm by dividing it into small parts to precisely find out the requirement of necessary factors like lime and other fertilizers and water based on co-ordinates of those areas.

- It's hard to inspect large-scale agricultural land manually and it's prone to human errors since we can't analyze these requirements precisely with manual inspection.

- Precision agriculture is the technique where you analyze satellite images of the farm and based on that you get to know about different factors like vegetation index, water clogging due to faults in irrigation system, presence of weed and pests present in different areas of your farm.

- Thus, using geographical information system (GIS) and GPS we can farmers to locate affected areas and take necessary action.

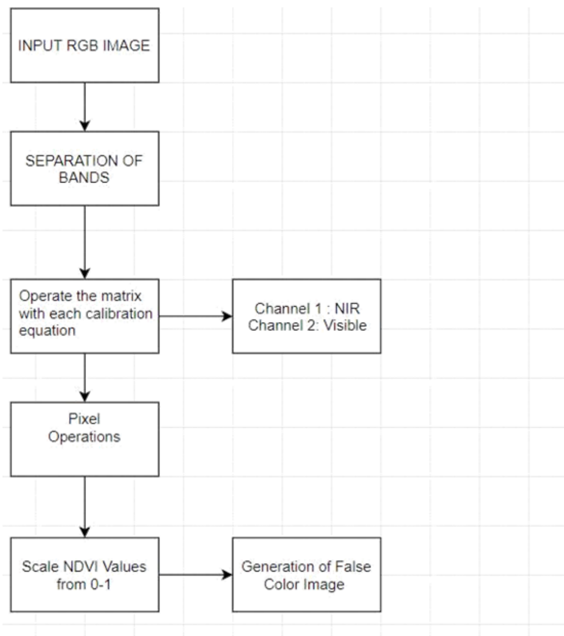
- Though these techniques were used since long time these are not precise since getting and processing images from satellites is time consuming and costly process. [1]

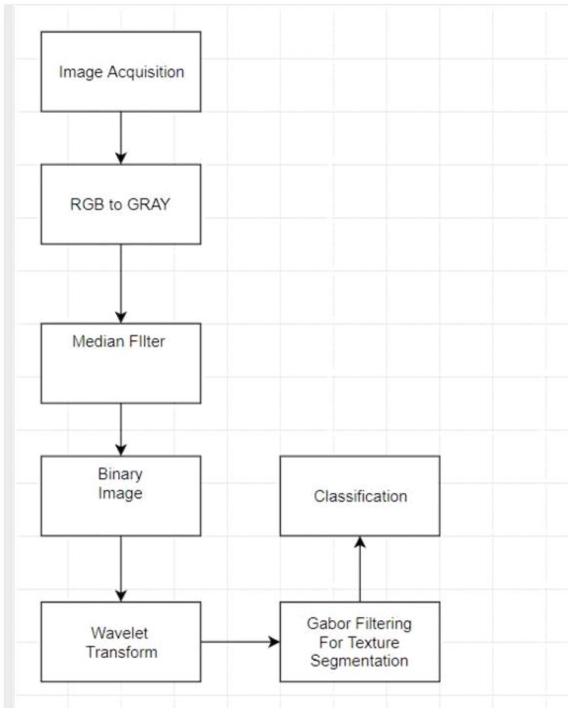
- Also, it’s not precise since it's taken from distance.

- Thus, to make this process efficient we would try to implement it using pictures taken by cameras mounted on farm monitoring UAVs

- Also, we will try to calculate get perpendicular vegetation index (PVI) instead of normalized difference vegetation index (NDVI) since it's more accurate measure of healthiness of the plant and which requires different image processing technique than what's traditionally used. [2]

Flowcharts:






**Question 4.**

**Option 1:**

Self-driving Car:

Minimum:

8. Lane detection, vehicle and pedestrian detection.
9. Road sign detection.
10. Speed control according to vehicle and road signs.

Target:

11. Steering wheel direction feedback according to the lane detection system.
12. Properly identifying the road signs and reacting appropriately according to identified signs.
13. Lane departure warning system.

Optimal:

14. Integrating google map APIs
15. Rear camera module to detect the vehicles behind and detect whether it's ok to change the lane.

**Option 2:**

Vehicle speed detection and license plate extraction using OpenCV

Minimum:

16. Keeping count of vehicles.
17. Tracking the speed of vehicles.

Target:

18. Capturing the image of the vehicle exceeding the speed limit.
19. Triggering the second camera to focus and capture number plate of the vehicle of interest.
20. Picking number plate region and extracting numbers and characters from number plate.

Optimal:

21. Use of character recognition and machine learning to exactly identify number plate.
22. Logging the number plate, speed of speeding vehicle along with date and time and store it in certain file or database.

**Option 3:**

Precision Agriculture using Image processing

Minimum:

23. Getting RGB images from satellite and using band separation and pixel manipulation techniques calculate approximate NDVI or use IR images and do the same to get accurate results. [1] [3] . Satellite options are AVHRR or Landsat 8.[5]
24. Feature extraction using NDVI to create false color composite image which would be spectral signature(geographical) of vegetation. [3]

Target:

25. Identify and locate water clogged areas using image segmentation and spectral analysis methods on resulting false color image.

26. Segregate different areas based on healthiness of the plants and identify their co-ordinates.
27. Differentiate between weed and plants using texture segmentation and classification based on training data-sets.

Optimal:

28. Image processing on images captured by drone instead of satellites which would increase the accuracy of the system, this would require stitching frames in the captured video and based on the path and speed of the drone and frame-rate of the camera to recreate the top view of the farm to do further image processing to get NDVI. (Drone Flying Site: Flatiron/Longmont Golf Course)

29. Write algorithm to calculate PVI instead of NDVI which would give more precise results.[2]

References:

30. https://www.agritechtomorrow.com/article/2018/01/ndvi-vs-false-ndvi--whats-better-for-analyzing-crop-health/10434

31. http://www.mdpi.com/2072-4292/2/3/673

32. https://ac.els-cdn.com/S2212017312006196/1-s2.0-S2212017312006196-main.pdf?_tid=90b17072-583d-4fbd-9a50-a61434a37da6&acdnat=1532834305_79e74a4c38bc4c0fa32af96c2a70f37f
33. http://www.microimages.com/sml/ParisScripts/PrecisionRemoteSensingSlides.pdf

34. http://proceedings.esri.com/library/userconf/proc17/papers/1981_658.pdf

35. http://www.magicandlove.com/blog/2011/12/04/people-detection-sample-from-opencv/

----