Jeevan Thomas Koshy

email: jeevan.thomaskoshy@gmail.com
Phone no: +91-9790469245
Github: https://github.com/jeevantk
Website: https://jeevantk.github.io/
TimeZone: +5:30 (IST)
Blog: jeevantk.blogspot.in

Face Alignment with OpenCV

Synopsis:

Face Alignment can be summarised as a process of finding keypoints in the image of a face and using those to find the exact orientation and pose of the face in the image. These detected points would be able to convey a lot of information like expressions, direction of view, etc. A major contribution in face alignment was done by Kazemi and Sullivan in their One Millisecond Face Alignment paper in 2014. This paper was published in the prestigious conference on CVPR 2014 and is still considered one of the best papers in the field. The algorithm requires the image of a face as an input which can be taken as an output of a generic face detector (for instance the Viola and Jones face detector).

Importance of this Project:

The first thought that comes to a layman's mind on hearing the word Computer Vision is that of Face Recognition and Detection. Viola and Jones made a significant contribution in detecting face from an image through their HAAR cascade classifier in 2004. This is still one of the most popularly used algorithms when it comes to face detection despite the advent of deep learning systems. OpenCV has an excellent implementation of this algorithm and is one of the most used algorithms in OpenCV. But when it comes to finding keypoints in the face for facial alignment the famous library is still a step back. There is an available implementation of the One Millisecond Paper on Dlib. This implementation was made around the time when the paper was published and that might be one of the reasons for OpenCV to not have an implementation of this paper. I personally felt the difficulty because of this when I was mentoring a project on Expression Morphing as a part of the Computer VIsion Group here at IIT Madras. You would be required to use two libraries in order to achieve a similar task i.e you need OpenCV for detecting the face and Dlib for computing the Keypoints in the face, which slows down development. Having a robust implementation of the Face Alignment paper can thus help the users to easily use implement these wonderful papers in their projects and create awesome products.

Deliverables:

I believe that an implementation alone would not create much impact if it is not accompanied with proper documentation as well as sufficient number of sample implementation of specific applications.The fact that there is a current implementation in Dlib will significantly help in the implementation process. I propose the following:

1.An OpenCV implementation of the Face Alignment [paper](#) by Kazemi and Sullivan in C++.
2.Proper Documentation of the implemented algorithm using Doxygen.
3.Using the above implementation for Expression Morphing.
4.Using above implementation for Face Swap and estimating the pose.
5.Creating OpenCV Python wrappers for the implementation.
6.Create a short Movie to demonstrate the code.

Project Description

The paper shows how an ensemble of regression trees can be used to estimate the face landmark positions from a sparse subset of pixel intensity values. Gradient Boosting is used for the actual learning of ensemble of regression trees.Gradient boosting works by creating an ensemble of a weak prediction models, usually decision trees. New trees grown to fit the residuals obtained as the error in current prediction by the ensemble with the target function. OpenCV already has Gradient Boosting inbuilt in it machine learning modules.But since we require a lot of modifications in the training steps it would be better to write our own Gradient tree boosting algorithm.

The paper takes into account two key elements that have been present in most of the successful algorithms in this space:
1.Reliable features is required for accurate estimation of shape and accurate shape is required for getting reliable features.An initial guess for the shape is made and an iterative approach is followed until convergence to find features and shape accurately.
2.The problem is difficult in the sense that it is non-convex(unlike a function with a single local minimum/maximum) and has multiple local optimums. The training can get stuck at one of the local optimas instead of reaching the global optimum. This problem is solved by assuming that estimated shape must lie in a linear subspace (computed from the principal components of the training shape.)
The major components of the implementation are:

1.Cascade of Regressors:
The shape is defined by a vector of points corresponding to the landmark points.The initial shape can be chosen as a mean shape of the training data centered and scaled according the detected face by a generic face detector(take Viola Jones Face detector as an example).Each regressor function makes an update on the current shape based on the image and the current shape estimation.This is process is done for all the regressors.

Each regressors are trained with gradient tree boosting algorithm.Further we do not require to enforce extra constraints since the outputs expanded by the ensemble is ensured to lie in a linear subspace of training data if the initial estimate belongs to this space and hence we are taking the average shape of the training data as the initial estimate.

2.Learning each regressor in the cascade:

The training data can be taken from the Helen Dataset which would give images and 194 corresponding landmark points in the face. The landmark points would constitute the shape vector . A triplet of images , an initial shape estimate and a target update step if formed for each regressors. The regression function is learnt using gradient tree boosting with a squared sum loss function.The set of training triplets are updated to provide the training data (the next triple) for the next regressor. This process is iterated a class of regressors are learnt which when combined gives a sufficient level of accuracy.

3.Tree Based Regressor:

Each regressor function is a tree based regressor fit to the residual targets during the gradient boosting algorithm.For a face with an arbitrary shape we want to find points which have same location relative to its shape as to the trained mean shape. This image can be wrapped into the mean shape based on the current estimate of the image. It would be computationally better if we only wrap the positions of the key-points instead of wrapping the entire Image. The wrapping can be approximated with the help of a translation, a scaling and a rotation matrix.

A regression tree approximates the underlying function as a piecewise constant function . Virtually we can generate any function as long as the tree is big enough.To start with we can generate a random set of leaf splits at each node and then we can choose those features which minimises the sum of square error greedily.

The decision at each node is based on thresholding the difference of intensity values at a pair of pixels.Even though this is simple this is much powerful than a single intensity thresholding.In order to account for the quadratic complexity of the problem we give preferential priority to nearby pixels by using an exponential prior.Using this reduced the prediction error on a number of datasets.

4.Handling Missing Labels:

This section describes how to handle issues such as incomplete labeling of training data. Even though we might not have this problem in our implementation it's still worth implementing.This can be solved by introducing an additional diagonal matrix whose elements can either be zero or one depending on where a landmark point is present or not. We can modify the gradient Boosting algorithm to account for these above changes and hence I suggest writing our own code for the gradient boosting for this paper's implementation.

The authors suggest 10 strong regressors each comprising of 500 weak regressors .The suggested depth of the tree is 5 . I would like to make an implementation with these values and also try to experiment to change the values in order figure out more details about the implementation.The authors have used HELEN Dataset for their primary training and testing purposes. I will also perform all the experiments mentioned in the paper to ascertain the validity of the implementation.Apart from HELEN Dataset they have also used LFPW for their training purposes. I would experiment over these datasets as well. I believe that an OpenCV implementation of this paper can help a lot of people both in Academia and Industry to develop cool Technological Solutions.

Making example Implementations:

I have proposed two main example implementation using the above implementation. Face Swap can be achieved by finding the landmark points,delaunay triangulation and by wrapping and blending the corresponding triangles.

Similarly expression can also be done by performing the required transformations on the source image to get to the target image on comparison with the landmark points position in both the images. Similarly facial pose can also be estimated as an approximate normal to all the landmark points. Computing facial pose can be very helpful in solving a problem of eye-gaze detection which has a lot of promising applications in many fields including security, advertising,etc. If time permits I would work on eye gaze detection as well after the finishing all the other deliverables.

Schedule

Our Institute has a three month summer vacation almost aligning itself with the GSOC period and hence I am totally free during the vacation.Our summer vacation commences on May 5th and ends on 31st July.GSOC schedule is from May 30th - August 21st. Even though I would be able to spend much time after the classes start I'm willing to start as soon as my summer vacation starts and finish the project before my institute reopens.

Timeline	Task	GSOC Timeline
May 8th-June 8th	Implement the entire paper in OpenCV	Community Bonding
June 8th-June 25th	Make proper Documentations using Doxygen and Implement the Python Wrappers.	Coding phase before first evaluations
June 25th-July 25th	Implement Face Swap,facial pose estimation and Expression Morphing with the existing implementations.	Coding Phase before Second Evaluations
July 25th-August 21th	Unit tests,Buffer time,Make example programs in Python using the wrappers created	Last coding Phase

Related Work:

There are two popular implementations available for the One Millisecond Face Alignment Paper.One of which is in the open source C++ library Dlib and the other is in DEST .Having two existing implementations makes the Job of implementation in OpenCV much easier. I have used the implementation available in Dlib earlier for Expression Morphing when I was mentoring a project as a part of the Computer Vision Group at IIT Madras.I am very comfortable with using OpenCV and have been using it for the last 3 years. I am comfortable in programming with C++, Python and Java.

Biographical Information:

I am a third year undergraduate student in the Department of Mechanical Engineering at IIT Madras. I developed an interest in Computer Vision due to the amazing Computer Vision Group here at IIT Madras.I was a part of the Group from my first year and is currently the Club Strategist. I have worked on various projects in Computer Vision and have Mentored a lot more projects done by Juniors.I have worked on extensively on Hand Gesture Recognition, Optical Music Recognition , Image Stitching ,3D reconstruction and Deep Learning.

I choose robotics as my Minor Stream and hence have done some courses related to Computer Vision and Machine learning. My relevant coursework includes Machine learning(from Coursera), Artificial Intelligence (CS6380) ,Computer Vision (CS6350) , Reinforcement Learning (CS6700) and Data Structures and Algorithms (EE4371). All courses apart from the Machine learning course were done from the IIT Madras.

I love teaching and regularly takes classes on Computer Vision as as part of the Computer Vision group.Regarding the implementation of the paper I am comfortable with almost everything I have mentioned above. I have not used Doxygen before but I am pretty confident that I can become familiar with it quite fast. I believe that I have a good understanding of the paper and am willing to read upon more if required.

I have interned with two startups before . In my second year I interned with a startup HyperVerge . This startup was founded by our former students of IIT Madras at the Computer Vision Group. Most of my time there was spend in learning and I trained a Convolutional Neural Network for classifying Documents vs Non - Documents. Last summer vacation I interned with a startup Detect Technologies. I was the lead Computer Vision Developer in this startup and I worked on processing drone Videos. I developed a Image stitching algorithm to stitch drone videos. Most of the ideas were taken from this paper by David Lowe. I was able to extend the stitching which was limited to rotational movements of the camera to translational motion as well (for objects at fixed depth).An output of this stitching is shown in my github page. I further worked on these problems and was able to come up with a novel way of 3D reconstruction . I was given a job offer in the startup for my contributions.

I would be available for full time for this summer vacation . Our Institute has a 3 month vacation which almost aligns with the GSOC schedule.My classes for the next semester will start on 31st of July. I won't be having a lot of courses next semester since I am going to be in my final year.Nevertheless I plan to complete the most of the work before that and hence I don't think that would be an issue. I am a music lover and plays piano in a nearby church which would require around 3 -4 hours of my time on Sundays. Apart from that I don't have any other commitments for this summer vacation and would be able to do justice to the project if I get selected.I am quite flexible with my sleep timings and would be able to attend all the IRC meetings .

I am applying for two projects this time in GSOC both under PSU. the other project is on Optical Music Recognition. I like both of these project topics but would give more weightage to the Optical Music Recognition Project since it would have a larger impact if completed.Thanks for taking the time to read my proposal. Hoping to work with Portland State University for GSOC 2017.

Extra Information:

- **Resume:** link
- **University Information:**
  - **University Name:** Indian Institute of Technology (IITM)
  - **Major:** Mechanical Engineering
  - **Minor:** Robotics
  - **Current Year:** Junior Year
  - **Expected Graduation Date:** June ,2018
  - **Degree:** Bachelor of Technology
- **Other Contact information:**

  - **Alternative Email:** me14b030@smail.iitm.ac.in
  - **Alternative Phone No:** +91-8547368410
- **Address:** Room No: 363,
  Alakananda Hostel,
  IIT Madras,
  Chennai,
  India.