

# Robust Real-time Head Pose Estimation for 10 Watt SBC

Emad Wassef<sup>1</sup>, Hossam E. Abd El Munim<sup>2</sup>  
Department of Computer and Systems  
Engineering, Ain Shams University,  
Cairo, Egypt

Sherif Hammad<sup>3</sup>, Maged Ghoneima<sup>4</sup>  
Department of Mechatronics Engineering,  
Ain Shams University,  
Cairo, Egypt

**Abstract**—Head Pose Estimation has always been an essential part for many applications such as autonomous driving and driving assist systems and hence performance optimization provides better performance as well as lower computing and power needs that allows us to run such applications over embedded devices inside these systems. In this article we present an implementation over a Single board computer for a new system of 3D Head pose estimation that estimates the Head pose of a person in real-time for applications such as Driver monitoring systems, Drones, Gesture recognition and tracking devices. The system is developed over a single board computer (SBC) that is suitable for very low powered applications, it only utilizes the data provided through the IR camera sensor to estimate both the Head and camera pose without any need for external sensors. This system will combine methods that include traditional image processing techniques for image projection, feature detection, key point description and 3D pose estimation along with Machine Learning techniques for face detection and facial landmarks detection.

**Keywords**—Head Pose Estimation; real-time; face detection; face landmarks localization; single board computing; SBC; GPU optimization

## I. INTRODUCTION

Realtime head pose estimation is a critical problem for many applications in the current industry. Applications such as autonomous driving where it provides assistance for the driver to monitor his attentiveness and drowsiness. This technique can also be used to provide real-time face recognition for tracking and monitoring systems on low cost boards, it can support future work in gesture and facial impressions recognition. In this approach we target the detection of head pose to support many applications like Driver monitoring systems, tracking drones, Gesture controls and augmented reality. This information is needed to detect the focus point and the concentration of the monitored person as well as to compensate for his motion to take the proper action accordingly. Our approach is to provide a system that is supposed to utilize low power profile consuming less than 10 watts while maintaining real-time operation.

As an example in a driver monitoring system, the driver assist camera need to check if the driver is concentrating on the road ahead or if he noticed the road signs and provide assistance and alerts, It can also support if he is trying to take a turn it can detect if he checked the side mirrors and that he is aware of the vehicles behind and his surroundings. Another example is in drone systems where it can be controlled by gestures and detect real gestures targeting the drone and not to be confused with fake gestures as part of his normal

behavior by detecting the pose is front facing the drone. It can be used for video surveillance systems where it's always tracking and centered over certain person and needs to provide a good follow up for his face.

Our proposed system is supposed to address these kind of applications taking into consideration the challenges imposed when using limited low cost hardware capabilities and provide low power consumption.

While several Head pose estimation techniques exists with very high accuracy like the state of the art Face Alignment technique of V. Kazemi[1], X. Zhu and D. Ramanan's method [2] and others, these methods among others all focus on the accuracy and discard other aspects like performance and efficiency, these two aspects depending on the application that the model is needed for can be considered as a limitation like in our situation for the use case of the use inside a very small single board computer board to be installed inside a low powered application. This is where our model comes in hand as we will discuss we can achieve higher computational efficiency from these models without sacrificing the accuracy.

There are several approaches and techniques to address the same problem, either by using full AI methods, Geometric methods, Tracking methods and lastly the Hybrid methods which combines multiple methods together.

In our system, we will follow the Hybrid method approach which is based on Geometric method with additional tracking algorithm. This shall provide robust pose estimation specially in case of tracking a moving person or using a movable camera. The tracking method will provide information that comes to hand to support in recovery from failure and in occlusion. Our proposed system will be based on a normal geometric method with a layer of Kalman filter to provide a tracking method on top of our geometric analysis.

In this paper we will present our proposed head pose estimation technique illustrated in Fig. 1 which consists of Face detection, Facial landmarks detection, Head Pose Estimation and Kalman filter. We will be discussing the problem definition of 3D head pose estimation in Section II followed by a survey analysis for the proposed techniques on the building blocks such as Face detection, Facial landmarks detection and Point to point mapping techniques as well as the different proposals and their challenges in the real-time manner showing the reliability of such system in Section III. Later on in Section IV we will discuss our new system and the hardware environment used in a production grade system specifying our selected

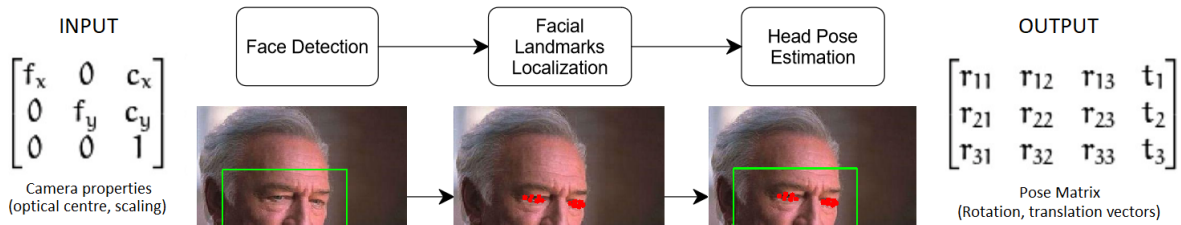


Fig. 1. A Block Diagram to Describe the Building Blocks of our Proposed System.

techniques and showing the relation between each phase with an illustrative algorithm and images for each phase. In Section V we will discuss our results and compare to the most famous state of the art techniques to provide a good insight of the performance and finally will concluded our research in Section VI.

## II. 3D HEAD POSE ESTIMATION

Pose estimation is a very popular topic in computer vision. It's problem definition is to find the extrinsic parameters of the camera matrix which are as following:

$$\begin{bmatrix} R_{3*3} & T_{3*1} \\ 0_{1*3} & 1 \end{bmatrix}_{4*4}$$

R & T represents the extrinsic parameters which transforms the coordinate system from 3D world coordinates to 3D camera coordinates. This means the Yaw, Pitch and Roll angles of the object in the 3D world coordinates system can be transformed to the camera coordinates system using linear equations give by:

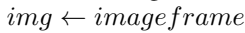
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R \begin{bmatrix} U \\ V \\ W \end{bmatrix} + t$$

It also incorporates the usage of a 3D model to map the 2D detected points on the image plan over a 3D plan so that we can get the U, V & W 3D world coordinates. Using this method to find the extrinsic parameters in real-time for human's head R and T matrices to estimate the pose. Our system needs to utilize the geometric method where it needs to find the 2D coordinates of specific points in any human's head. This means we need to find the face from within the image frame and then find the pre define points in this image frame. These preselected points are needed to be mapped to the 3D model using point to point perspective for a generic Human head. In order to do so, the system needs first to do Face Detection, followed by Landmarks localization for these pre selected points, given these points we shall use PnP and solve for Pose Estimation, this workflow can be illustrated from Algorithm 1. Other methods are also applicable to combine steps together to find the 2D coordinate of these points.

### Algorithm 1 Head Pose Estimation Geometric Method

---

**Require:** Camera Intrinsic parameters *Cam\_Mat*  
**Require:** Face Detector Model *FD\_model*  
**Require:** Facial Landmark detection Model *FL\_model*  
**Require:** Head Model Reprojection Matrix *RP\_Mat*

**while** New image frame **do**  
     ← *imageframe*  
    *Faces* ← *FD\_model*(*img*)  
    **if** *Length*(*Faces*) > 0 **then**  
        **for** *Face* in *Faces* **do**  
            *Landmarks* ← *FL\_model*(*Face*)  
            *SolvePnP*(*RP\_Mat*, *Landmarks*, *Cam\_Mat*)  
            *R\_vec*, *T\_vec* ← *SolvePnP*  
            *R\_Mat* = *Rodrigues*(*R\_vec*)  
            *Pose\_Mat*[*Face*] ← *R\_Mat*, *T\_vec*  
        **end for**  
    **end if**  
**end while**

---

## III. REALTIME HEAD POSE ESTIMATION CANDIDATE TECHNIQUES

While our primary goal is to find a robust yet real-time solution to the Head pose estimation problem based on the Geometric approach we selected, several techniques were suggested to achieve the best computational performance without sacrificing accuracy. The geometric approach consist of three main parts that may be combined in technique consists of a single model or the general technique of a standalone model for each stage, these stages are as follows:

- 1) Face Detection Technique.
- 2) Facial Landmarks Localization Technique.
- 3) Point to Point perspective for Head pose estimation.

### A. Face Detection Technique

Here we discuss different techniques for face detection from within the image, the purpose of this step is to identify the face and return a bounding box identifying the location of the face within the image as illustrated in Fig. 2. If a face is detected, it's location will further on be supplied to the Facial landmarks detector for further processing, most of face detection approaches are based on machine learning models that are trained on object detection.

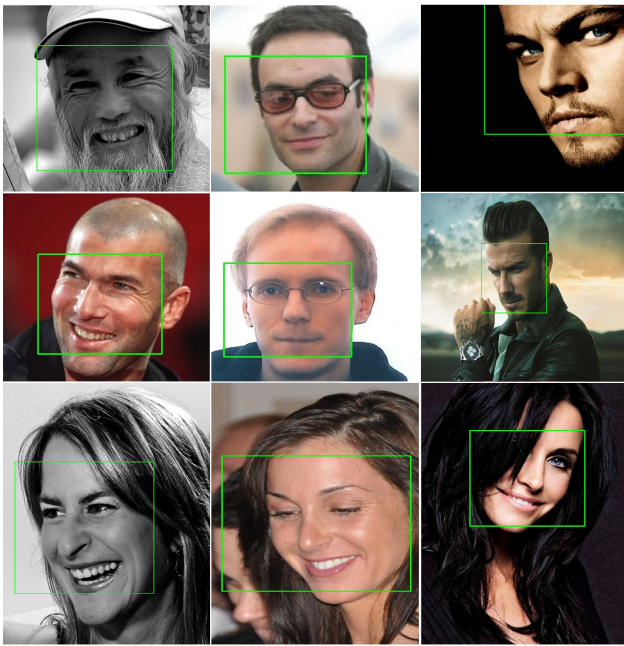


Fig. 2. A Face Detection Sample of the Hybrid Algorithm using MMOD Technique[6].

The first proposed technique was the well known Haar feature-based cascade classifiers which was proposed by Viola and Jones [3][4], this technique consists of three main stages Haar features acting as the convolution kernel and plenty of these features are calculated, then selection process over these feature to provide with relevant and informative features and discard the weak features using adaptive boosting technique. The classifier is the combination of the strong features. This technique provides good accuracy and has good computing efficiency but suffers greatly under occlusion and non-frontal faces.

Another proposed technique is using the Histogram of Oriented Gradients, the HOG method for face detection is based on the same concept of object detection as depicted by the well known pedestrian detection technique 'Histograms of oriented gradients for human detection' [5], the HOG descriptions using sliding windows are extracted for positive and negative samples then used for training a linear SVM, of course the sliding windows and multiscale windows are used to detect objects within the image frame regardless of their location and scale then we suppress the non-maxima bounding boxes as we will end up having multiple detection for the same object. While HOG method provides better performance than the Haar cascade classifiers method in terms of occlusions, non-frontal facing and illumination invariance, it's still computationally expensive as sliding windows techniques are usually expensive specially when we are doing operations like gradient calculations in HOG yet still faster in and offers superior performance than that of the previous Haar cascade classifiers.

Another technique like Max-Margin object detection [6] for HOG CNN based features was taken into consideration, This technique is known to be slower than the HOG method but provides superior performance in facial detection in cases

of occlusion and hard detection cases.

While there are many more techniques that contributes to the same issue like TinyFaces, DSFD [7] and ASFD and many others that perform great in terms of multiple faces detection with various benchmarking datasets like WIDER [8] and Fddb [9] yet this increased accuracy for detection of multiple faces is irrelevant to our application as we only need to detect single face, also they are much more computational intensive than that of our system can't provide. These previous methods are concerned for face detection only, there are also techniques that can do extra task, in particular some techniques can do both the face detection as well as landmarks localization, these techniques shall be taken into consideration against the previously suggested alternatives for face detection integrated with the face landmark detection techniques altogether for fair judgment, for this we will discussing them in the next section with face landmark detection.

### B. Facial Landmarks Localization Technique

Face landmarks detection is the last step before Head pose estimation to detect the preselected points in Human's face, it starts by providing the algorithm with location of the face within the image frame bounded by a box as in Fig. 2, the model then processes this bounding box to provide the location of some points in the face as in Fig. 3 where we can identify the eye, eyebrows, nose, mouth and jawline if needed. There are many datasets that differentiates between the facial landmarks. Our selected dataset was based on iBUG dataset "a semi-automatic methodology for facial landmark annotation" [10], [11], [12], this base dataset contains 68 landmark points and it's created from several other datasets including the LFPW [13], HELEN [14] and other datasets which results in 7674 images in total, these images were annotated using a semi-automated annotation process, this provides a vast number of training and testing data that are very rich in variations in conditions and imposes more challenging situations for the proposed techniques.

Another notable mention is the HELEN dataset [14] that originally provides 194 localization points as facial landmarks instead. It consists of 2000 images for training and 330 for testing. Usually more localization landmark points translates to better accuracy, but also this comes at the cost of computational needs during technique's inference operation in runtime. Among the vast proposals of face landmarks localization techniques we have the ensemble of regression trees V. Kazemi and J. Sullivan's technique [1], this technique was claimed to have one millisecond performance per image, this technique provides good benchmarks compared to many state of the art techniques, we put this among our proposed solution for this reason however, they didn't mention the Hardware used to benchmark their claimed computational needs. Another technique was proposed by S. Ren LBF technique [15] which claims to reach 3000 fps while providing better results by utilizing the concept of local binary features search instead of using a global one over the whole image frame which provides better error rate when validated over the two datasets of our choice. Other techniques that can do multiple operations like the one introduced by Multi-task Cascaded Convolutional Neural Networks MTCNN technique[16], this technique can directly identify the face and also do five point facial landmarks





Fig. 3. Facial Landmarks Localization Output using our Proposed Method.

detection, this technique stands out in real life results for both face detection and Landmarks localization but it's five landmark points only which will not provide enough information for applications such as drowsiness behavior or emotion state, it's also performing much worse in terms of computational needs as it provides less than one fps thus can't be utilized for real-time operations for low power applications, another suggested technique is proposed by Zhu and Ramanan [2], this technique does all, the face detection, landmark localization and the pose estimation which is quite very useful as it sums up all the stages for pose estimation, yet still it has huge cost in terms of computational need as they claim it takes 40 seconds per image to provide high accuracy compared to the commercial existing solutions.

### C. Point to Point perspective for Head Pose Estimation

Lastly, Point to Point perspective needed to calculate head pose estimation. At first we need to compensate for scale variation by using the size of the face box according to the face detection algorithm, afterwards we need to have an estimated 3D model for the face in order to map each facial 2D point to it's corresponding 3D face model, this mapping is needed to find the Head Pose from this 2D-3D correspondence. Many proposed methods exist for such called perspective point problem (PNP), one of the methods is based on Levenberg-Marquardt optimization another which provides higher efficiency is the Effective PNP method [17] and lastly is a method using the direct least square solutions for PNP [18] which provides better accuracy and error rates without sacrificing real-time operation than the previous two methods. Other approaches exist that rely on noise free points Like P3P and P4P approaches but they proven not suitable for our application as they provide inaccurate results when there are higher noise in these selected points. After estimating the Pose, we apply Kalman filter for further filtering the output in order

to have a smooth transitions and better correlation between each frame and the preceding which reflects real life behaviors.

## IV. SINGLE BOARD IMPLEMENTATION FOR 3D HEAD POSE ESTIMATION

### A. Our Proposed System

Aiming to target real-time performance on a production ready solution we selected a well known and stable platform that supports low power applications with industry grade performance and capabilities that also supports easy expandability. Our Selected control board is Jetson Nano board provided by Nvidia depicted in Fig. 4. This system specifications are illustrated in Table I. It's constructed of a Quad-core ARM Cortex-A57 MPCore processor along with NVIDIA Maxwell GPU with 128 NVIDIA CUDA cores to support image manipulation and model inference functions included in our algorithm. This single board computer (SBC) module consumes less than ten Watts with all peripherals connected. This board is 69.6 mm x 45 mm in size which is smaller than a credit card in size which makes it very suitable in small low powered applications, to have imagination about how low is it's power consumption, When installed in any product, Our system's power needs is less than the tenth of the power needed by the smallest motor installed in a small drone, it's almost the same power consumed by a toy DC motor. This proves how efficient it can be used in low-powered embedded applications.

TABLE I. NVIDIA JETSON NANO MAIN SYSTEM SPECIFICATIONS.

CPU	Quad Core ARM Cortex A-57 1.43GHz
GPU	128 Maxwell Cores
Memory	4 GB 64 bit LPDDR4 25.6 GB/s
Storage	16GB eMMC
Camera	12 MIPI CSI-2 DPHY 1.1 lanes 1.5Gbps
Peripherals	PCIe USB3.0 SDIO SPI SysIO GPIOs I2C
Power Consumption	5 Watt or 10 Watt modes

This board runs Linux operating system environment, this allows us to use many libraries such as OPENCV and DLIB[19], also models that are built using Caffe, Pytorch or TensorFlow. This is amazing feature to run a lot of variations as well as building our custom models.

The rest of our system is constructed of a single IR camera sensor accompanied by IR LED lights to provide good illumination for the image and to prevent distraction and confusion with no additional sensors are used in our system. This camera sensor supports frames of resolution up to 1,920x1,080 for 30 frames per second with scalable viewing angle and focus. This system is supposed to provide the rotation matrix parameters Yaw, Pitch and Roll for both the head pose in case of movement. The camera is assumed to have a constrained range of motion around all three-axes assuming that no sudden far jumps can happen so that there will always be a correlation between a frame and it's preceding one.

As illustrated earlier, Our system shall consist of several stages as shown in Fig. 1, Each stage contribute to our problem definition as shown in Algorithm 1. We expect that the human



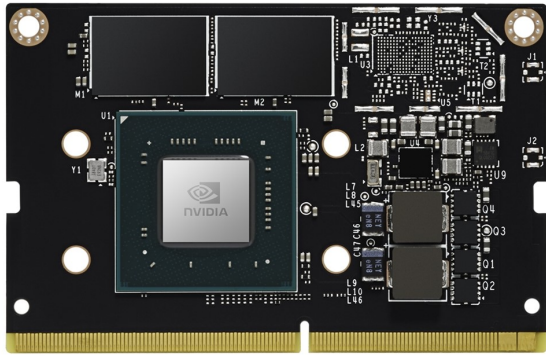


Fig. 4. NVIDIA Jetson Nano Production Module.

will have a front facing camera with a constrained movement that majorly will be facial facing but encounter change in yaw, pitch and roll angles. Our system will perform all the stages on video stream in real-time manner to provide robust tracking of the Head's pose using the Hardware we presented. We split the system into these three main stages where we can address each at a time:

- 1) Initialization.
- 2) Our Head Pose Estimation System.
- 3) Tracking Kalman Filter.

### B. Initialization

As mentioned earlier our system consists of a single camera sensor, this sensor is like any camera needs to be calibrated, this calibration process is needed to find the intrinsic parameters of the camera sensor which is done using checkerboard method [20], the checkerboard method measures the intrinsic parameters in terms of focal length and the principal point, also it measures the extrinsic parameters, these data is used to build the calibration matrix needed later on in both the Camera and Head pose estimation, the extrinsic parameters defines the 3-D location of the camera with respect to the world coordinate which are used to translate from the 3-D camera coordinate system to world coordinate system, the intrinsic parameters is used in the transformation from 2-D coordinate system in the image to the 3-D camera coordinate system, lastly camera distortion matrix can also be calculated which can be used in correcting image distortions like radial and tangential distortions but they are not mandatory. This step is done for a single camera sensor at the first setup and these values later on to be used as standard configuration for the system, they need to be done for autofocus sensors and provide some kind of look up table or through equation as these parameters are dependent on the focal point.

### C. Our Head Pose Estimation System

Geometric method of head pose estimation as we discussed earlier constitutes of 3 main parts, Here we will state our selected techniques from the ones we suggested earlier and in Section V we shall see comparison of it's performance and how our selected techniques compares to others.

Using a real-time video stream, we shall feed our face detection algorithm on a frame by frame basis based on Max-

Margin object detection [6] method, This method provides more robust results than the other two methods in field tests,

This method is supposed to be slower than HOG method however, using GPU accelerated implementation of this method it performed great in our test that somehow surpassed the other approaches due to it's higher accuracy. MMOD method provides two points representing the bounding box for the head location which is suitable to be provided to our own custom model of head landmarks detection needed for face alignment problem.

Max-Margin Object detector technique is a generic technique that's trained over face detection dataset Fddb [9], the MMOD uses similar approach to HOG in building HOG descriptors using sliding window classification method and utilizing random projection based locality sensitive hash for determining which bin the calculated HOG descriptor belongs to, this projection process is less in computational cost than the original HOG method, yet the whole process is more expensive due to the sliding window operation. This method was selected to elevate the accuracy of face detection as it's the corner stone of the Head pose estimation to correctly identify the head from within each image frame.

In the other hand, for the facial landmarks localization technique, we build our own custom model to be used which is based on the ensemble of regression trees similar to V. Kazemi's [1] but with a modification in the set of facial landmark points, this approach was done over three phases to get the best computational efficiency without sacrificing the accuracy. The first phase is to select feature points that gives us relevant information regarding our needed aspects as shown in Fig. 5 where it illustrates the most important landmarks that provides comparable results according to 68 Landmarks Dataset, Fig. 5b shows our selected points are those that describes eye lid opening, head orientation and mouth contour, which are the points that contributes mostly to the drowsiness behavior, emotional state, and has higher impact on determining face orientation. Secondly analyzing the impact of each model parameter on the overall performance of the model and selecting the optimum range for these parameters. The last phase is to do parameter optimization over 150 combinations of the selected ranges for each model parameter and calculate metrics for overall accuracy over both train and test data, training and inference speed and lastly size, these metrics are used for comparative analysis with the original models as well as other alternative solutions.

Lastly, Point to point perspective, we used the direct least square PNP [18] which provides both accuracy and real-time operation. This function is already implemented in OPENCV library and can be used directly by providing it the Model points and the corresponding points from the previous step of Facial landmarks localization. Solving this PnP problem we get our  $R_{vec}$  and  $T_{vec}$  which constitutes the pose matrix

### D. Tracking Kalman Filter

Kalman filter named after Rudolf E. Kálmán is a modeling technique used for predicting the next state from the history of previous data. This technique is widely used in cases of trajectory and motion planning as well as signal processing algorithms as it provides robustness to statistical noises and

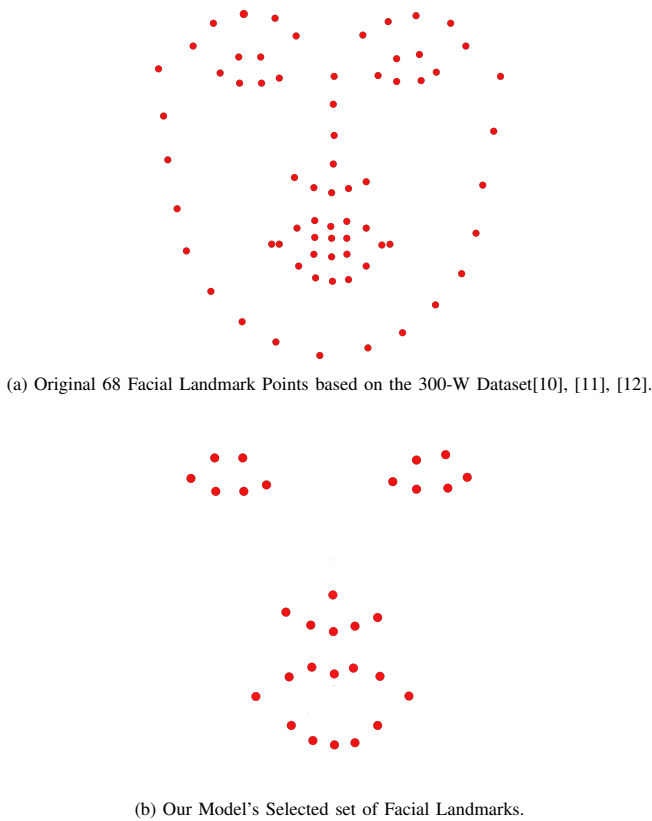


Fig. 5. Our Proposed Selected Set of Facial Landmark Points Compared to the Original 68 Landmark Points[10], [11], [12].

sudden abrupt changes that are not following the trajectory pass according to the previous state. In it's simplest form, it follows linear equations however, this filter contains many complex variations and can predict the state of 2D, 3D and more complex non linear equations states with many variations like Extended Kalman filter, unscented, Hybrid and many other variations. Here in our applications we used the simple form of Kalman filter to predict the pose estimation to provide it as the simplest for of tracking algorithm to the head pose. We firstly start by constructing the Kalman filter and then by starting the application we feed the filter with data after each phase until the filter converges and then it shall provide filtration from occlusion and drop or incorrect data during operation for sudden short periods of time.

## V. EXPERIMENTAL RESULTS

In our experimental test we will show how each of the Face Detection and Facial Landmarks localization techniques perform in respect of Realtime operation and accuracy of detection. Also we will be showing the Benchmark results of them compared to each other in terms of error rate to show how accurate each technique shall perform.

For the Face detection operation we will be comparing between Viola and Jones technique [3] [4], HOG technique[5] and Max-Margin object detection [6] technique in terms of the speed of inference and error rate as depicted in Fig. 6. Our diagram shows how each technique compares to each other

in terms of speed and accuracy of detection. Our accuracy of detection method is according to Fddb metrics of the ROC curves based on continuous score method for the True positive rate at 1000 False positive samples threshold. Our results indicate that the FaceDetection process is the bottle neck of the whole algorithm as the frame rate are much lower than the other parts of the algorithm.

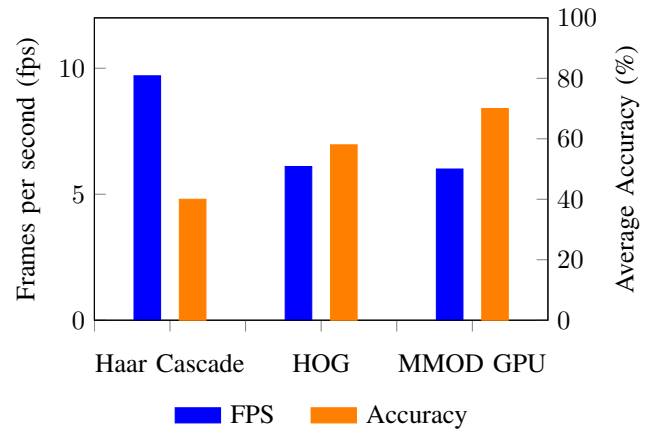


Fig. 6. Comparison between the three Proposed State of the Art Face Detection Techniques in terms of Accuracy and Performance as Frames Per Second.

In terms of Face landmarks localization, in Fig. 7 we show how our reduced feature points technique performance is compared to the original S. Ren LBF technique and V. Kazemi ERT technique using our hardware configuration. We will apply same approach of V. Kazemi's technique where only 31 points are selected instead of the full 68 landmark points of the 300-W dataset in order to maximize speed without sacrificing accuracy.

This illustrated accuracy is calculated using 1000 images from testing dataset in addition to 6674 from the training dataset of combined datasets including iBUG dataset [10], [11], [12], LFPW [13] and HELEN [14] for testing purpose using the same criteria as described by V. Kazemi. Calculation of the accuracy was done using their provided function of DLib library, this function calculates the distance of each landmark to the ground truth position, this value is divided by the interocular distance to get the normalized distance of the landmark. Using the average of this normalized distance we can get the error and hence the accuracy as depicted in our diagram. Our technique shows very good results compared to other techniques while on a larger dataset with much more sparse training and testing data. As illustrated in Fig. 8, We showed a sample of images from the three datasets we used AFW, HELEN, IBUG and LFPW where multiple faces are detected correctly with different poses and lighting conditions as well as immunity to different occlusion types and glasses wearing yet there are certain scenarios that the system fails in either identifying the face or to locate the landmarks correctly, this normally happens in very minor situations when it's in a very acute pose, or when the occlusion is severe that eliminates crucial parts of the face and makes it hard to detect the distinctive facial features.

Also using AFLW2000-3D dataset [21] which is 2000

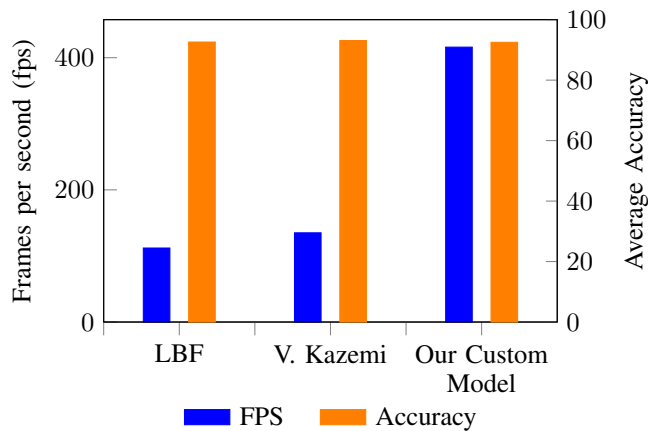


Fig. 7. Comparison of Face Landmarks Localization Techniques between our Custom Model Versus the well known Models of LBF [15] and V. Kazemi [1] in Terms of Performance and Accuracy of Detection.

image can be used for 3D face alignment evaluation, our system which is based on DLib approach with reduced number of points showed Mean average error of 16 degrees versus 9 of the Ground Truth landmarks.

Our benchmarking tests of the full system performs with total frame rate around five frames per second for sampling a 640x480 monochrome image which is very good performance. This is done utilizing only the GPU of our evaluation board, leaving the Quad-core ARM Cortex-A57 almost free for other application threads that will utilize the output of this algorithm. This leaves more room for further parallelization, optimization and off loading to the Quad-Core CPU if needed.

## VI. CONCLUSION

We presented a real-time full system for head pose estimation for embedded infrared camera sensor, that's capable of compensating for the motion of the camera and provide the 3D Head pose estimation accordingly. We presented our new system including the techniques of Face detection, Facial landmarks localization and Point to point mapping that are optimized for operation over embedded low powered devices that has limited computational abilities. We also presented the challenges in each step as well as the possible techniques that are suitable for our applications presenting our own modification for the Face landmark localization technique showing both speed and accuracy of operation. Lastly we presented the pose estimation calculations and Kalman filter for smoothing out fluctuations and better tracking. As for the future direction, Our model can be enhanced to use the detected facial landmarks of the human and analyzes his drowsiness and detect this facial emotions and provide further more information beyond the pose estimation.

## REFERENCES

[1] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[2] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec 2001, pp. I–I.

[4] —, "Robust real-time face detection," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 747–747.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[6] D. King, "Max-margin object detection," *ArXiv*, vol. abs/1502.00046, 2015.

[7] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang, "Dsf: Dual shot face detector," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5055–5064.

[8] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5525–5533.

[9] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[10] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: database and results," *Image and Vision Computing*, vol. 47, pp. 3–18, 2016, 300-W, the First Automatic Facial Landmark Detection in-the-Wild Challenge. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0262885616000147>

[11] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 896–903.

[12] —, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[13] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[14] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 679–692.

[15] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692.

[16] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, pp. 1499–1503, 2016.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epn: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, 02 2009.

[18] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *2011 International Conference on Computer Vision*, 2011, pp. 383–390.

[19] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[20] Zhengyou Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, 1999, pp. 666–673 vol.1.

[21] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2016, pp. 146–155. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.23>



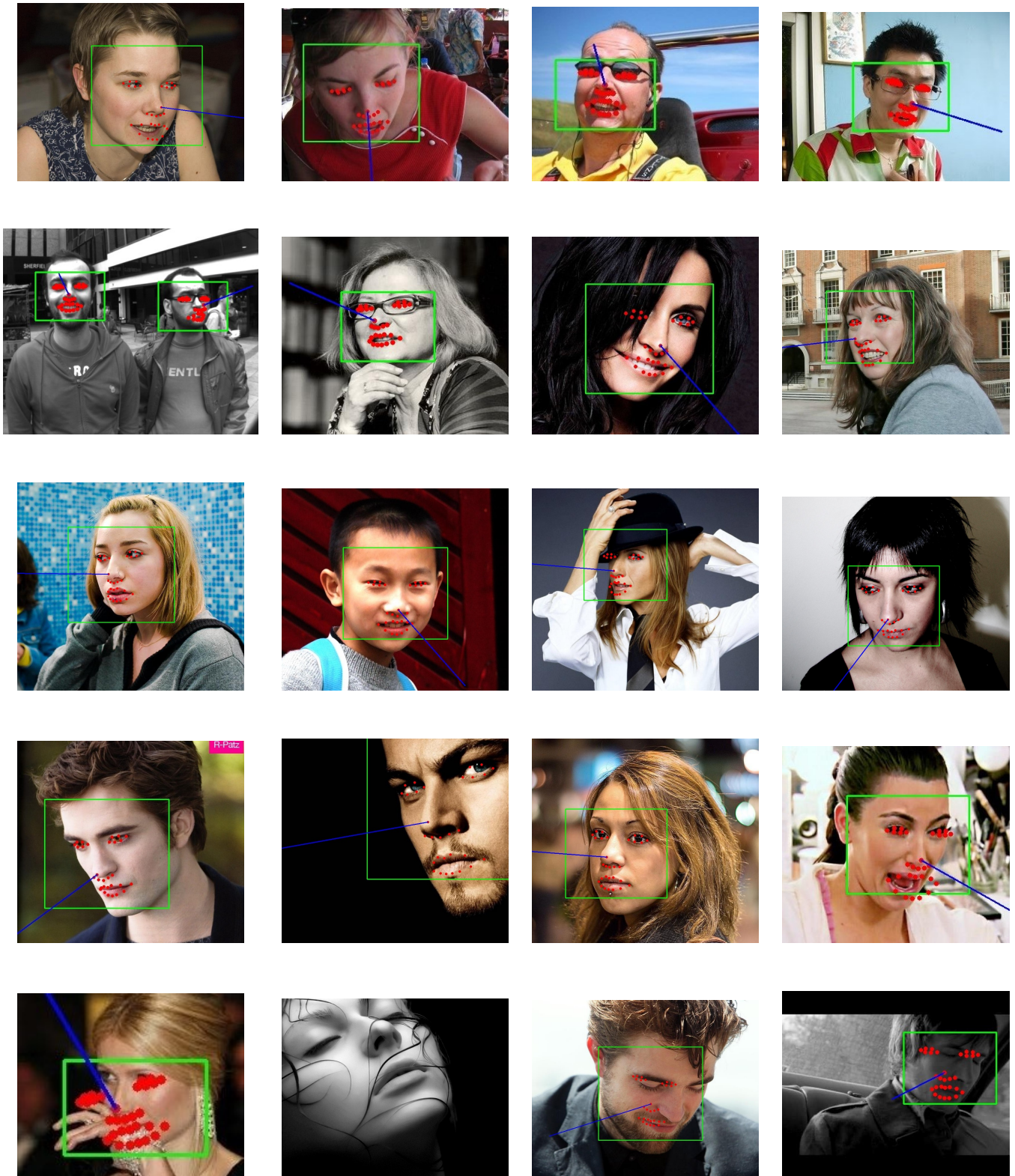


Fig. 8. Sample of the Output from our used Datasets to show Immunity to Partial Occlusion, Lighting Conditions, Wearing of Glasses and Different Poses with the Last set Showing Scenarios of a Negative Result.