

ActiveNet: A computer-vision based approach to determine lethargy

Aitik Gupta
ABV-IIITM, Gwalior
aitikgupta@gmail.com

Aadit Agarwal
ABV-IIITM, Gwalior
agarwal.aadit99@gmail.com

ABSTRACT

The outbreak of COVID-19 has forced everyone to stay indoors, fabricating a significant drop in physical activeness. Our work is constructed upon the idea to formulate a backbone mechanism, to detect levels of activeness in real-time, using a single monocular image of a target person. The scope can be generalized under many applications, be it in an interview, online classes, security surveillance, et cetera.

We propose a Computer Vision based multi-stage approach, wherein the pose of a person is first detected, encoded with a novel approach, and then assessed by a classical machine learning algorithm to determine the level of activeness. An alerting system is wrapped around the approach to provide a solution to inhibit lethargy by sending notification alerts to individuals involved.

KEYWORDS

Computer Vision, Human Pose Estimation, Pose Encoding

ACM Reference Format:

Aitik Gupta and Aadit Agarwal. 2021. ActiveNet: A computer-vision based approach to determine lethargy. In *8th ACM IKDD CODS and 26th COMAD (CODS COMAD 2021)*, January 2–4, 2021, Bangalore, India. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3430984.3430986>

1 INTRODUCTION

Activeness, both physical and mental, has been one of the primary healthcare concerns since the inception of smart devices. It has undergone a significant drop, especially since the COVID-19 outbreak [11]. This is due to lazy and sedentary practices, either during leisure time or while working from home.

Despite recent advancements in Computer Vision and related domains, reliable security surveillance systems with little to no human intervention [8] is still a challenge, especially now that these times call for unfortunate motivations and impulses.

There has been significant research on drowsiness detection using Computer Vision [1], but most of them leverage the degree to which the person's eyes are open or closed. While this setting is instrumental for in-car webcams, they fail for long-range distances, which is a standard paradigm in CCTV footages, security webcams, et cetera.

Moreover, numerous papers related to the study of body language using Computer Vision pertain to emotion detection [12]; wherein there has been an apparent *lack of attention, given to attention*.

With the outlook of a more generalized solution, we demonstrate a multi-stage mechanism to identify activeness of a person, such as in online classes, job interviews, security surveillance systems, et cetera, aiming to rectify diminished activeness in students, interviewees, security guards respectively. The input to the pipeline would just be a single RGB image. To realize lethargy in this type of multi-stage approach, an intermediate representation of information is essential. At the end of the first stage, we maintain a 2-dimensional Cartesian plane coordinate information of various joints of a person. Just using this representation, it is almost impractical to achieve our goal, which drives the idea of our novel pose encoding stage, where we maintain an angular representation of data, aiming to remove all positional aspects in the data. With abstraction after every stage, the resulting system becomes more robust to the inherent *noise* of the data, such as visual characteristics of various people, different coordinates for different image views or sizes, et cetera.

2 ARCHITECTURE

ActiveNet is a machine-learning based system, the proposed architecture for which accepts a camera input. It is then operated through a Human Pose Estimation module to extract keypoints of the joints in human body. The pose encoding module generates a vector, which is calculated by the angles of certain joints. The angles are imputed and scaled via intermediaries generated while training a self-scraped dataset, discussed later. After processing, a traditional machine-learning classifier takes the angles as input. The classifier predicts the activeness level, based on which the alert system is triggered.

2.1 Human Pose Estimation

2.1.1 Literature Review. Human Pose Estimation (HPE) is one of the key aspects in computer vision that has undergone tremendous research in the last few years. Its numerous applications are one of the main reasons for the importance it has gained. It estimates the configuration of the body (pose) from a single, typically monocular, image. To study the methods of estimating the position of joints from an image, one must understand some of the most significant challenges:

- Background noise, lighting, visibility issues
- High variance of visual appearance and physique of humans
- Partial occlusions due to self-articulation and layering of objects
- Complexity of human skeletal structure and its hierarchy
- Information loss by projecting a 3D object to a 2D plane

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CODS COMAD 2021, January 2–4, 2021, Bangalore, India

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8817-7/21/01.

<https://doi.org/10.1145/3430984.3430986>

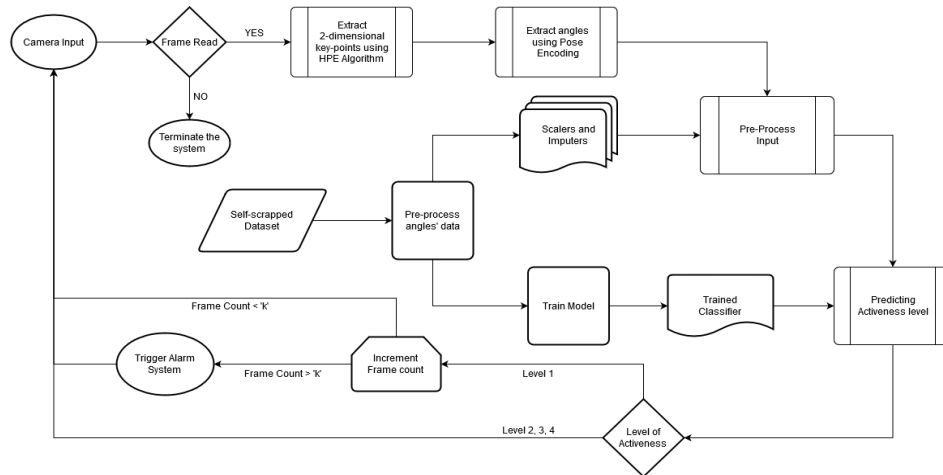


Figure 1: ActiveNet Architecture

Some classical approaches are based on a pictorial-structure framework. Early works introduced a mixture model of parts, which expresses joint relationships [19]. This approach has the limitation of having the pose model independent of image data. Later on, deep learning based regression approaches were introduced [17], which brought a shift in the research paradigm towards them. Most of the later researches operate over convolutional building blocks, and have been universally adopted for image-data driven approaches. Gradually, direct keypoint regression-based methods were replaced by more promising heatmap regression methods [18].

Broadly classified, there are two approaches to the convolutional architectures for Single Person Pose Estimation (SPPE) or Multi-Person Pose Estimation (MPPE). The first approach, the **top-down** approach, is decoupled into two sub-problems. Firstly, a person detection algorithm is claimed, followed by a pose estimation algorithm for every detected person. State-of-the-art (SOTA) solutions for the sub-problems could potentially be utilized together in the pipeline. The inference speed of this approach strongly depends on the number of detected people inside an image. The second approach, called the **bottom-up** approach, is more resilient to the number of people in an image, and thereby, could potentially be faster than the first approach. Firstly, all possible keypoints are detected in an image, followed by grouping by human instances.

2.1.2 Our Work. We based our work on the popular bottom-up method Lightweight OpenPose [13], for mainly two reasons. Firstly, this work heavily optimizes the original OpenPose [3] implementation to reach real-time inference speeds on CPU with negligible accuracy drop, which can further be optimized for Edge Devices using Intel® OpenVINO™ Toolkit, as done in other researches [9]. Secondly, in the context of finding the levels of alertness in a person, our work is dependent on the angles generated by different body joints. Therefore, the **bottom-up** approach is well suited for isolation of joints, even in cases where other parts are not detected. We propose a pre-processing step for the pose encoding, which takes care of the joints for which HPE module could not determine the 2D locations of keypoints. It will be discussed later in the paper.

We leverage the pre-trained weights provided by open-sourced library at: <https://github.com/Daniil-Osokin/lightweight-human-pose-estimation.pytorch>, which contains the implementation of Lightweight OpenPose [13]. The weights are trained over the COCO dataset [10], consisting of 17 keypoints in the following order: (1) Nose; (2) Neck; (3) Right Shoulder; (4) Right Elbow; (5) Right Wrist; (6) Left Shoulder; (7) Left Elbow; (8) Left Wrist; (9) Right Hip; (10) Right Knee; (11) Right Ankle; (12) Left Hip; (13) Left Knee; (14) Left Ankle; (15) Right Eye; (16) Left Eye; (17) Right Ear; (18) Left Ear. The Neck (2) keypoint is just a 2-dimensional mean of Right Shoulder (3) and Left Shoulder (6) keypoints in the Cartesian coordinate system, therefore making it a total of 17 keypoints, along with one additional extrapolated keypoint.

2.2 Pose Encoding

We propose a pose encoding technique, with the aim of removing any positional configurations from the detected keypoints. We claim this by taking the angles between different subsets of joints in the upper and lower body. Regardless of the position of a person in an image, the absolute values of angles between the joints remain same for a single type of pose. We do not use the keypoints directly, as Figure 2 explains that even if the images are just a mirror of each other and the contextual pose of the person remains same, their estimations can be quite different. Those are dependent on the positional aspects in the camera input. A slight change in the camera’s offset would lead to an entirely different set of 2D keypoint estimations. Another motivation for angular representation can be explained by Figure 3; as the angle between shoulders, neck and head changes, the abstract activeness of the person changes along with it. We consider the angles made by following subset of joints:

- (1) Nose, Eyes, Ears
- (2) Neck, Nose, Ears
- (3) Shoulders, Neck, Nose
- (4) Neck, Shoulders, Elbows
- (5) Shoulders, Elbows, Wrists
- (6) Nose, Neck, Core Joint
- (7) Neck, Core Joint, Hips

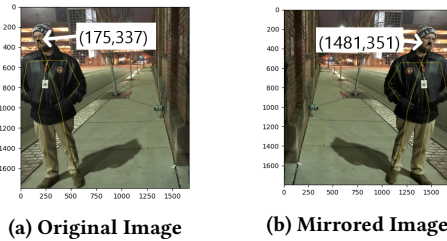


Figure 2: Nose keypoint estimation with augmentation

(8) Hips, Knees, Ankles

The subset is chosen such that it captures the possible range of motions in most sections of human body, and thus, numerous poses. We consider Core Joint as the 2D mean of Left Hip and Right Hip joints in the Cartesian plane, and the actual angles are calculated using dot product.

Encoding a single pose creates an array of 15 elements, containing angular data in degrees. The joints which are not detected in the pose, are represented by (-1,-1) in the coordinate axes. We give NaN values in the encoding when two out of three keypoints (from **A**, **B** and **C**) have the same 2D coordinates, which includes the (-1,-1) case. This can well occur due to occlusions in the image, or semi-accurate estimations. If this is not handled during encoding, a **ZeroDivisionError** error is raised. NaN values are later handled in subsection 2.4. Since this module inputs just 2D coordinates of joints detected by HPE algorithm discussed in subsection 2.1, it is invariant to the visual aspects of the image, inherently expanding the generalizability to new image-data.

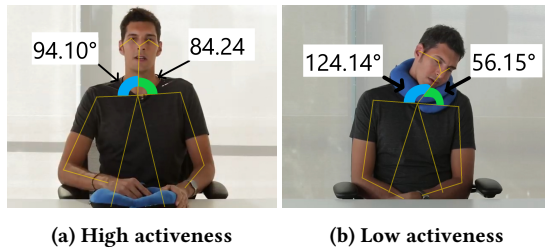


Figure 3: Angular Encoding

2.3 Dataset

To train our model on the encoding from the module discussed in subsection 2.2, we scraped images from web using different keywords for different classes. The dataset contains 4 classes for 4 levels of activeness-zones:

- (1) Level 1: Below 25%
- (2) Level 2: Between 25-50%
- (3) Level 3: Between 50-75%
- (4) Level 4: Above 75%

There are 40 images for each class, scraped using keywords such as "army soldiers" for Level 4 activeness, while "sleeping while standing" for level 1 activeness.

2.4 Pre-Processing

2.4.1 Treating NaN Values. Pose encoding yields NaN values as discussed in subsection 2.2. Treating these values is a prominent

discussion in data science. One approach is to eliminate the encoding which has more than half NaN values. Another approach is to use numpy.ma module [7] which provides a convenient way to address this issue, by introducing masked arrays, which are either no-mask representing only clean entries or boolean arrays indicating presence of invalid entries, in which case the invalid entries are eliminated, allowing the classifier to train on only valid entries. The approach of pose encoding as described in subsection 2.2 and as shown in Figure 3 makes this possible, even with fewer features.

2.4.2 Scaling Features. The raw features from the encodings are scaled down to a similar range across the whole dataset. This helps in many ways during training, as priority can be given to optimizing the weights based on the correlation of features to the target rather than scales of various features. We use the standard scaler provided by scikit-learn for this module.

2.5 Training

To train a classifier using the scaled data discussed in subsection 2.4, we adapt the ensemble algorithms based on Decision Trees [14], such as Random Forest Classifier (RFC) [2] and XGBoost Classifier [4]. We first train and validate a Logistic Regression algorithm [15] to establish the benchmark scoring for classification. Subsequently we tune the hyper-parameters using scikit-learn's GridSearchCV. The best scoring results are then chosen for each algorithm, while using K-Fold cross-validation technique. Table 1 provides algorithm-wise results.

2.6 Evaluation and Analysis

Table 1: Classifier-Wise Results

Classifier	Accuracy
Logistic Regression	56.67%
Decision Tree Classifier	66.67%
XGBoost Classifier	63.34%
Random Forest Classifier (RFC)	76.67%

Table 2: Class-Wise Results for RFC

Class	Precision	Recall	F1-Score
Level 1	0.989	0.820	0.896
Level 2	1.000	0.745	0.853
Level 3	0.914	0.663	0.768
Level 4	0.989	0.807	0.888

Based on experimentation, the best results are achieved using RFC with an accuracy of 76.67%. As Figure 3 explains, even a small subset of joints can be adequate to assess the overall activeness. Hence a general interpretation can be lined that tree-based ensemble algorithms would do better here, as the individual trees are built on subsets of features. A detailed evaluation of the classifier results is given in Table 2. Analysing the results, we find that the extreme classes have the best scores compared to the intermediate classes.

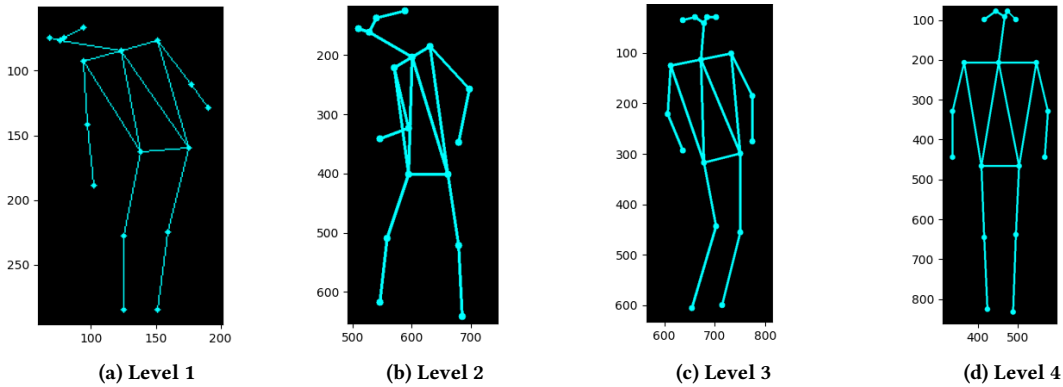


Figure 4: Activeness Levels

Our self-scraped dataset does not represent an ideal distribution of the real-world classes, the outcomes so achieved only provide a baseline solution for our approach.

2.7 Alert Mechanism

At the time of inference, an alerting mechanism provides a method to inform people in-concern if diminished levels of activeness is developed in the targeted person. This is done in faith to keep the targeted person agile. To get rid of false positives, and in order to make the pipeline more robust, we raise a notification only when k number of contiguous frames are classified in the lowest class, where k is an arbitrary constant. A reasonable value of k can change with the domain. The alert module relies on Slack Workspaces. Other works have used the same utility, but for different use cases [6] [5].

We configure a *notification-alert bot* after enabling the **Incoming Webhook** functionality in Slack Workspace:

- (1) Webhook is a unique URL to send HTTP requests
- (2) Only organizations with the Webhook URL can send alerts
- (3) Only organizations in the workspace can receive alerts
- (4) The alert can be sent with a customized message, along with a time-stamp
- (5) Works on all platforms, requires Slack to be installed

For demonstration purposes, we create a demo Slack Workspace **active-networkspace**, typically meant for an organization. Here for example, it is meant for *active-net* organization.

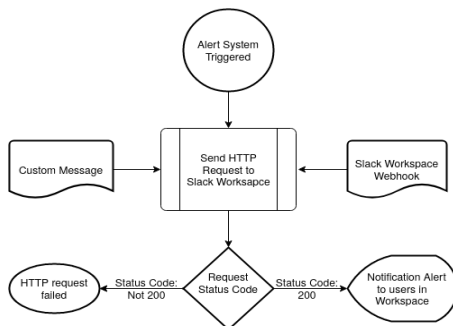


Figure 5: Alert Mechanism Flowchart

3 DEMONSTRATION

For demonstration, we use a desktop webcam, rigged with a single Nvidia GeForce GTX 1650 graphics card to run the whole pipeline, inferencing at around 34 frames per second. The code, along with screenshots are available at: github.com/aaditagarwal/ActiveNet

4 CURRENT LIMITATIONS

As there exists no well defined dataset for such type of classification task, in addition to discussion in subsection 2.6, we end up with weights which do not generalize for *unseen poses*. One of the main complication arises when camera view angle shifts. For instance, a situation where the person is looking sideways, the keypoint estimations will be from a totally different distribution compared to estimations when person is looking directly at the camera. Consecutively, angles between joints do not provide a good encoding solution in this situation. Considering the limitations of the self-scraped dataset, we overlooked the implications of such issues.

Another limitation to be realized is the loss of information projecting a 3-dimensional object to a 2-dimensional space. The 2D keypoint estimations, and their corresponding encoding, both suffer from this limitation. Considering the recent developments in 3D pose estimation, which are either end-to-end [16], i.e. operate on a monocular image, or take 2D estimations from HPE modules and lift them to 3 dimensional space [20], theoretically, encodings could be improved further to support angles made in 3 dimensions. Given our aim for a real-time solution, we decided not to address the 3 dimensional approach, but can be approached in future.

5 CONCLUSION

Through this paper, we explained our multi-stage approach to identify levels of activeness with a novel pose encoding stage. Once trained on enough poses, it can potentially be used in any domain without retraining, given that the HPE module is robust enough to generate the keypoints. We demonstrated the idea with a working pipeline, using Slack to alert users in a workspace with custom messages. We welcome future research in this domain with a strong belief that **Activeness should be given active attention**, both mental and physical.

REFERENCES

- [1] M. A. Assari and M. Rahmati. 2011. Driver drowsiness detection using face expression recognition. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. 337–341.
- [2] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1.
- [4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco, California, USA) (KDD '16)*. ACM, New York, NY, USA, 785–794. <https://doi.org/10.1145/2939672.2939785>
- [5] Aidan Connolly. 2018. Automated Outlier Detection in Crime Data Using Programming. *Undergraduate Honors Thesis, University of Nebraska-Lincoln* (2018).
- [6] Brandin Grindstaff, Makenzie E. Mabry, Paul D. Blischak, Micheal Quinn, and J. Chris Pires. 2019. Affordable remote monitoring of plant growth in facilities using Raspberry Pi computers. *Applications in Plant Sciences* 7, 8 (2019), e11280. <https://doi.org/10.1002/aps.3.11280> arXiv:<https://bsapubs.onlinelibrary.wiley.com/doi/pdf/10.1002/aps.3.11280>
- [7] UHC Group. [n.d.]. Missing data: masked arrays¶. https://currents.soest.hawaii.edu/ocn_data_analysis/_static/masked_arrays.html
- [8] T. Ko. 2008. A survey on behavior analysis in video surveillance for homeland security applications. In *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*. 1–8.
- [9] Endah Kristiani, Chao-Tung Yang, and Chin-Yin Huang. 2020. iSEC: An Optimized Deep Learning Model for Image Classification on Edge Computing. *IEEE Access* PP (02 2020), 1–1. <https://doi.org/10.1109/ACCESS.2020.2971566>
- [10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 740–755.
- [11] Warwick J McKibbin and Roshen Fernando. 2020. The global macroeconomic impacts of COVID-19: Seven scenarios. (2020).
- [12] Fatemeh Noroozi, Ciprian Căneanu, Dorota Kamińska, Tomasz Sapiński, Sergio Escalera, and Gholamreza Anbarjafari. 2018. Survey on Emotional Body Gesture Recognition. *IEEE Transactions on Affective Computing* PP (01 2018). <https://doi.org/10.1109/TAFFC.2018.2874986>
- [13] Daniil Osokin. 2018. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. In *arXiv preprint arXiv:1811.12004*.
- [14] Vili Podgorelec, Peter Kokol, Bruno Stiglic, and Ivan Rozman. 2002. Decision Trees: An Overview and Their Use in Medicine. *J. Med. Syst.* 26, 5 (Oct. 2002), 445–463. <https://doi.org/10.1023/A:1016409317640>
- [15] Claude Sammut and Geoffrey I. Webb (Eds.). 2010. *Logistic Regression*. Springer US, Boston, MA, 631–631. https://doi.org/10.1007/978-0-387-30164-8_493
- [16] Jun Sun, Mantao Wang, Xin Zhao, and Dejun Zhang. 2020. Multi-View Pose Generator Based on Deep Learning for Monocular 3D Human Pose Estimation. *Symmetry* 12, 7 (2020), 1116.
- [17] Alexander Toshev and Christian Szegedy. 2014. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [18] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. 2016. Convolutional Pose Machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4724–4732.
- [19] Yi Yang and D. Ramanan. 2013. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 35, 12 (dec 2013), 2878–2890. <https://doi.org/10.1109/TPAMI.2012.261>
- [20] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. 2019. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3425–3435.