Improving Post-Processing Methods on Video Object Recognition Using Inertial Measurement Unit

Zhiyuan Zhou
Brown University
zhiyuan_zhou1@brown.edu

Abstract

This paper considers two post-processing models based on Inertial Measurement Unit (IMU) to enhance the accuracy of video object recognition on light-weight devices. Videos are rich with temporal information, and IMU is a cheap and accurate way of accessing it. The work combines temporal information of IMU with two post-processing models: 1) Intersection over Union model and 2) Kalman Filter, both of which require small memory and low compute time, making them an ideal choice for light-weight device. A video data set with IMU data is collected and processed using a popular classification CNN You Only Look Once (YOLO). The recognition results are then passes on to the above two models, and the results are compared to show that IMU data can significantly increase recognition accuracy on both models.

Index terms – Inertial Measurement Unit, video object recognition, post-processing, Intersection over Union, Kalman Filter

1. Introduction

In recent years, object recognition on image has achieved great success. Well-trained Convolutional Neural Networks (CNNs) can successfully detect objects in a single RGB image and classify them with a high accuracy. However, video object recognition is still a hard problem because of motion blur, occlusion, etc. Recent approaches in dealing with recognition on video mostly runs image object recognition algorithm on every frame of the video. However, this could potentially be a very time-consuming task, and can't always operate in real time. More importantly, it doesn't take into account the inherent temporal information of videos.

This paper proposes to capture that temporal information with an Inertial Measurement Unit (IMU), a small electronic device that capture's an object's 3-axis linear acceleration and 3-axis angular velocity, and sometimes device orientation as well as magnetic field strength. In a video, the underlying temporal information suggests that objects in one frame is most likely to be also in the next, with some pixel movements. Using an IMU, one can use the camera's movement to predict how the objects will move in the video, providing extra information to strengthen object recognition results.

To make this work applicable for all object recognition networks, the model is made a post-processing one, which can take the result of any recognition network and strengthen it. Two primary models are considered for this post-processing unit: 1) Intersection over Union (IoU) model and 2) Kalman Filter (KF). IMU information is incorporated with these two well-established models. IoU is a geometrical way of calculating how "close" two detected objects are, and IMU data is used to determine whether two close objects detected from adjacent frames are the same one in the real world. The recognition confidence of a repeatedly seen object is bolstered over time. Kalman Filter, on the other hand, is a Markov model that uses observations over time to smooth noise and inaccuracies. It is used in this paper to capture the temporal information of videos to take away abrupt changes in recognition confidences. A data set of video is collected on a fish eye camera with its corresponding IMU measurements. Since many of the current image object recognition networks take a good deal of computing time and compute power and can't operate on every frame of a video in real time, this work focuses on the post-processing work of video segmented into 1 frame per second. The low frame rate enables real-time video object recognition even on devices with low compute power. The 1 fps video is first processed by a popular object detection frame work You Only Look Once (YOLO) [12], then passed on to the post-processing model. The results of the post-processing model is then compared to results from YOLO, showing a significant increase in accuracy. The use of IMU make the result of a low frame rate video comparable to that of a high frame rate one. It is also shown that IoU operates better in smooth stable videos, while KF is better in dealing with sudden changes in videos.

The remaining sections of this paper are organized as

follows: Section 2 surveys related work on video object recognition post-processing methods and on IMU usage. Section 3 describes the Intersection Over Union model and Kalman Filter model, and how they can be combines with IMU data. Section 4 presents the experiment methods and data. Section 5 focuses on the experiment results, as well as review and comparison of the two models. Section 6 serves as the conclusion of the paper and points out direction for future work. The code base of this paper is available at: https://github.com/paulzhou69/object-recognition-imu

2. Related Work

The author did not find any related work done on using the Inertial Measurement Unit (IMU) for the object recognition problem. However, much work has been done on improving object recognition with Kalman Filters / Hidden Markov Models (HMMs) and the Intersection over Union technique.

Inertial Measurement Unit There has also been use of IMU in problems of 3D reconstruction and feature matching [9], recognizing gestures [16], and body poses [4]. However, this work is different from these previous work in that it focuses on using the IMU specifically for the object recognition problem.

Hidden Markov Models Kubala [8] has proposed a HMM for temporal smoothing task for hand positions. Many other works has also been done using HMM for problems such as classification [2] [11] [4] and pattern recognition [5] [15]. Hidden Markov Models are effective in these problems because it assumes a stochastic model of the world where some information aren't directly observable. Hornegger *et al*. [7] have already attempted to solve the object recognition problem using HMM combined with affine invariant geometrical features. Bicego *et al*. [3] have also proposes an HMM-based approach to deal with appearance-based 3D object recognition. This work is different in that it considers the 2D object recognition problem using a specific kind of HMM: Kalman Filter.

Kalman Filter KF is really successful at combining different sources of data to make a more accurate estimate of its real value. Besides estimating the value, it also keeps track of its covariance matrix to represent its believability. Alatise and Hancke [1] has developped a method to apply Kalman Filter to IMU and vision data to estimate the pose of a robot. The Kalman Filter model in this work operates in a similar way but is used to solve the object recognition problem. Rong *et al*. [13] has explored using an HMM to regulate the Kalman Gain in Kalman Filters. The method has shown promising results, but is outside the scope of this work, which only considers a regular Kalman Filter with unregulated gains.

Intersection over Union IoU is traditionally a way of evaluating the performance of a object recognition network. However, Han *et al*. [6] used it as a medium to find high-scoring detections from nearby frames to boost those with a low score. Although their method was effective in improving the performance in object recognition, it cannot operate in real time. It also didn't seek to make use of IMU data to obtain a more accurate IoU score. In contrast, this work presents an IMU-based IoU model that only looks at the previous one frame in real time to boost detections with low confidence.

3. Proposed Models: IoU and KF

3.1. Inertial Measurement Unit

Inertial Measurement Unit (IMU) is a small electronic device that includes a gyroscope and accelerometer, and sometimes a magnetometer. In this work, only the gyroscope and accelerometer data (3-axis angular velocity and linear acceleration) is used.

By amounting an IMU to the camera, the movement of the camera can be fully described by the IMU data. Since the time between frame is relatively small, we can assume that objects in the camera are still in the real world, and the objects' relative movement to the camera is exactly the reverse of the camera's movement as obtained from IMU. This relative movement data is available for every frame in the video and can be used to calculate the displacement of objects detected between two adjacent frames.

3.2. Intersection over Union

Intersection over Union, hereafter known as IoU, is a commonly used metric of object detectors. Recent object detector can generate a rectangular bounding box around a detected object, and IoU is a mathematical formulation of calculating how close two bounding boxes are. IoU of box A and box B is computed as follows:

$$IoU(A, B) = \frac{Area(A \cap B)}{Area(A \cup B)} \tag{1}$$

If IoU score is above 0.6, we say that two bounding boxes are sufficiently close to each other.

In every frame, let $B(t) = \{b_{t1}, b_{t2}, b_{t3}, \dots\}$ be the set of bounding boxes detected at frame t. Each bounding box b_{ti} is a list of full probability distribution over all the classes of objects the recognition network can recognize, where each probability O_j is a tuple with the confidence b_{ti} contains the jth class object, the x,y coordinates of the center of the rectangular bounding box, and the width and height of the box.

$$b_{ti} = [O_{t1}, O_{t2}, O_{t3}, \dots, O_{tN}] \tag{2}$$

$$O_{tk} = (c_{tk}, (x_{ti}, y_{ti}, w_{ti}, h_{ti})) \tag{3}$$

where N is the total number of classes of objects recognizable to the network. The object in bounding box b_{ti} is

eventually defined to be in class k with probability c_{tk} if and only if $c_{tk} = max\{c_{t1}, c_{t2}, \ldots, c_{tN}\}$

At frame t, the IMU measurement I_t can be used to calculate the displacement dx, dy as described in Section 3.1. In turn, the two displacement is applied to every class probability O_{tk} of all b_{ti} in $B(t)$:

$$O'_k = (c_k, (x_i + dx, y_i + dy, w_i, h_i)), \forall k, i \tag{4}$$

This yields a new set of bounding boxes, $B'(t)$, after adjusting for displacement. $B'(t)$ is actually a prediction of $B(t+1)$ using $B(t)$ and IMU data.

This prediction can be used to check the actual results $B(t+1)$ and, if the two correspond, enhance the reliability of the result. For each bounding box $b'_{ti} \in B'(t)$ and $b_{(t+1)j} \in B(t+1)$, if

$$IoU(b'_{ti}, b_{(t+1)j}) > 0.6 \tag{5}$$

$c_{(t+1)l}$ of $O_{(t+1)l}$ in $b_{(t+1)j}$ is increased, where l is the object class of b'_{ti}. After the increase, bounding box $b_{(t+1)j}$ is defined to be in object class k iff $c_{(t+1)k} = max\{c_{(t+1)1}, c_{(t+1)2}, \ldots, c_{(t+1)N}\}$

This increase process in then repeated for every frame t, resulting in a sequence of increased detection result.

3.3. Kalman Filer

Kalman Filter (KF) is a special kind of Hidden Markov Model (HMM) that is specially designed to take in a series of noisy measurement over time and smooth the noise and inaccuracies to output more accurate results. It repeats the process of "predict" and "update" on the input data. For a specific data point, the "prediction" phase uses that data and the specified transition dynamic to predict the data at the next time step. The "update" step then updates the real data at the next time step with the predicted one, essentially taking a weighted average of the two.

The underlying hidden state of the system is represented by a vector x:

$$x = [O_{1a}, O_{1b}, O_{2a}, O_{2b}, \ldots, O_{Na}, O_{Nb}]^\top \tag{6}$$

where each O_{ia}, O_{ib} is the probability of object class i as defined in (3), and N is the total number of classes of objects recognizable to the network. Each object class occurs twice in the state vector (with subscripts a and b) to allow for any object class to appear for a maximum of two times in a frame. If an object class is detected more than 2 times in a frame, only the two with the highest confidence is incorporated in the state vector, while others are passed directly to the output without going through the KF model. This limit of maximum occurrence for each object class contribute to a relatively small requirement of compute power and processing time, while remaining highly effective for common videos. Alongside x, the filter also keep a covariance matrix P at every time step to represent the correctness of x

The transition dynamic of the state vector is modeled by a transition matrix $F_{10N*10N}$:

$$F = \begin{bmatrix} 1.2 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & & & & \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \tag{7}$$

$$\text{where } F_{ii} = \begin{cases} 1.2 & \text{if } i \equiv 1 \mod 5 \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

Each index in x that represents confidences are multiplied by 1.2 while all other indices stay the same. All the confidence are increased by 20% at each time step since we have reason to believe that an object detected in one frame is very likely to also be present in the next frame. In addition, x is also affected by the input u_{4N*1} at each time step and a control matrix B_{10N*4N}:

$$u = [dx_{1a}, dy_{1a}, dx_{1b}, dy_{1b}, dx_{2a}, \ldots, dx_{Nb}, dy_{Nb}]^\top \tag{9}$$

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \ldots & 0 \\ \ldots & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \tag{10}$$

$$\text{where } B_{ij} = \begin{cases} 1 & \text{if } j = 5\lfloor \frac{i+1}{2} \rfloor - 2 - (i \bmod 2) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Each dx_i, dy_i is the displacement calculated in Section 3.1 for each class probability O_i.

The prediction step of the KF is modeled by the following equations:

$$x = Fx + Bu \tag{12}$$

$$P = FPF^\top + Q \tag{13}$$

Q, the process noise covariance matrix is the identity $I_{10N*10N}$

The update phase is done by taking a weighted average of the prediction of x and the actual observation using Kalman Gain K as the weight:

$$S = HPH^\top + R \tag{14}$$

$$K = PH^\top S^{-1} \tag{15}$$

$$x = x + K(z - Hx) \tag{16}$$

The measurement function matrix$H$ and observation noise covariance matrix R are both identity matrix $I_{10N*10N}$, and z is the observation vector.

Figure 1. An example image from the video data self-recorded by fish eye camera. YOLOv4 is able to recognize the classes "person", "backpack", and various others in this picture.

By repeating the predict and update step at every frame of the video, the Kalman Filter uses the temporal information to increase the confidence of objects correctly detected and lower those of falsely detected.

4. Experiments

The two models described in Section 3 are tested and evaluated using the ImageNet ILSVRC2015 data set [14], a data set by Ovrén and Forssén [10], and self-collected data.

A wide-angle fish eye camera (159° horizontal 127° vertical angle of view) is used to record a video of an indoor environment, which is then segmented into 1 frame per second images (See Figure 1 for an example). Adafruit BNO055, a 9 degree-of-freedom IMU, is used to record orientation, velocity, and acceleration data. The data is then processed on a Nvidia RTX2060 GPU using the popular object recognition network YOLOv4 and the two post-processing models.

The detection results, the confidence of the objects detected and their bounding box location, are recorded after the images have been processed by YOLO. The results are then passed to the Kalman Filter and Intersection over Union post-processing model separately, and both of the final result are also recorded.

5. Results: Comparison and Analysis

The results of the self-collected data set processed by YOLO, YOLO + Kalman Filter model and YOLO + Intersection over Union model are saved respectively and compared.

Performance of object recognition task is usually evaluated by its Average Precision (AP) score. The AP score shows how a object recognition model does in terms of its ability to pick up true positive items and disregard false positives and false negatives. In this section, performance of the two post-processing model is evaluated by AP score, recognition confidences at every frame for every object in video, and bounding box location of detected objects. In this section, the AP score is calculated with regard to the ground truth in a subset of the ImageNet ILSVRC2015 data set [14].

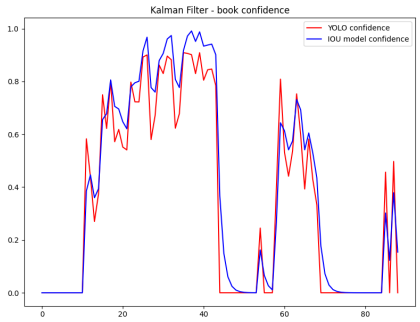

Figure 2. Recognition confidence of YOLO and the KF model of a "book" in the video sequence. KF output confidence, in blue, outperforms YOLO output confidence, in red, when "book" is detected in some number of consecutive frames. KF output confidence also increase sudden drops in confidence that YOLO incorrectly predicts.

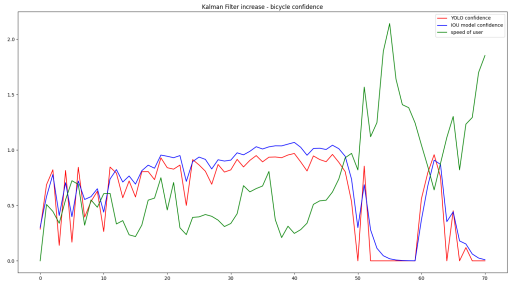

Figure 3. Recognition confidence of YOLO and the KF model of a "bicycle" in the video sequence is in red and blue respectively, and the speed of the camera is recorded in green. Recognition confidence is very low for both models when camera speed is large.

The compute time of these two models are also recorded and compared. Overall, both of the models operate sufficiently fast to process low frame rate videos in real time. The Intersection over Union model is especially fast, running at an average of 0.002 seconds per frame, while the Kalman Filter runs at an average of 0.1 seconds per frame.

5.1. Kalman Filter model results

The performance of KF is much similar to that of a noise filter. It can effectively reduce the variance of recognition confidences, so that sudden changes in recognition confidence, such as those caused by noise and inaccuracies, will be smoothed out. This is illustrated in the two example Figure 3 and Figure 2 above.

The bounding box location of YOLO and that of the Kalman Filter model is also compared. It is shown that when camera speed is relatively small, the two locations are

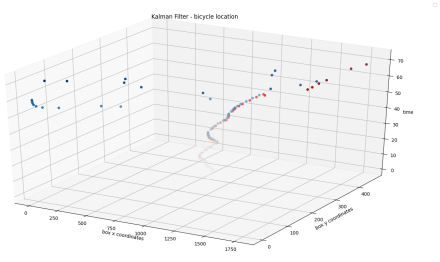

Figure 4. Location of bounding box of YOLO and KF. The xy plane represents the x,y coordinates of the center of the bounding box, and the z-axis is the frame number in the video. YOLO bounding box location, in red, is close to the Kalman Filter bounding box location, in blue, in the lower half when camera speed is low, but diverge in the other half when camera speed greatly increases.

Video	YOLO AP	Kalman Filter AP
1	1.00	1.00
2	0.85	0.84
3	0.48	0.50
4	0.49	0.49
5	0.99	1.00
6	1.00	1.00
7	0.98	0.99
8	0.98	0.98
9	0.82	0.77
10	0.94	0.94
11	0.75	0.75
12	0.81	0.70
13	0.67	0.67
14	1.00	1.00
15	1.00	1.00

Table 1. Results. Ours is better.

close to each other, but vary greatly when camera moves rapidly, as shown in Figure 4

Finally, the performance of Kalman Filter is evaluated formally using Average Precision (AP). The AP score of the Kalman Filter is almost always about the same as the AP score of YOLO, with 1% fluctuations (See table 1). Moreover, the change in AP score is not affected by the movement speed of the object in the video.

This AP score data shows that the Kalman Filter model doesn't outperform the original object-recognizer in terms of getting true positives and rejecting false positives. However, it is very effective in smoothing the confidence of the object detected when there is much noise in recognition confidence, and in increasing the confidence of the detected object when the recognition confidence is already smooth. The noise-canceling property of Kalman Filter also makes

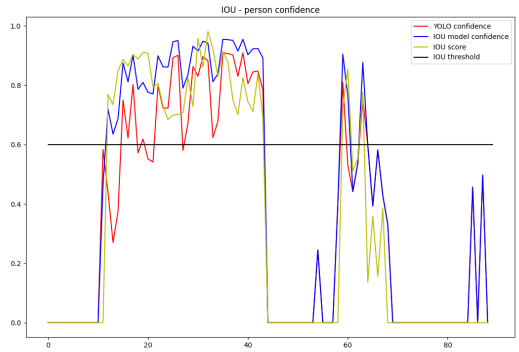

Figure 5. The IoU model with IMU recognizing a "person" in a video. The yellow line shows the IoU score at every frame. When the IoU score is above the threshold (represented by the blue line), the confidence of the object gets increased from the original YOLO output confidence in red to that in blue.

it especially effective in dealing with objects that moves at a high speed.

5.2. Intersection over Union model

The IoU model can effectively use bounding box locations to trace how the object moved throughout the entire video. Because of this quality, it is very successful in using information from the previous frame to boost detected object confidences in the current frame. As in figure 5, the IoU model is very effective at increasing the recognition confidence of objects.

The data also shows that the Initial Measurement Unit (IMU) greatly increases the IoU score for 1 frame per second video sequences. For example, comparing the IoU score in figure 5 (with IMU data) and figure 6 (no IMU data, pure IoU model) shows an average increase of 50%, with maximum increase at 150%. The increase in IoU score immediately leads to a significant increase in the recognition confidences.

By the design of the IoU model, its bounding box location is the same as the detection algorithm's (YOLO's). Therefore there is no change in the accuracy of bounding box location.

Finally, the Intersection over Union model is formally evaluated using again Average Precision scores. It is shown that, similar to the Kalman Filter, the IoU model neither increase nor decrease the AP score by more than 1% most of the time. However, there are times where a huge increase/decrease up to 30% - 40% is seen (see table 2

However, it is found that the change in AP score is correlated with the movement speed of the detected object with regards to the video frame. When the object is barely moving, the AP score is more likely to decrease. When the object is moving in the frame, however, the AP scores tends

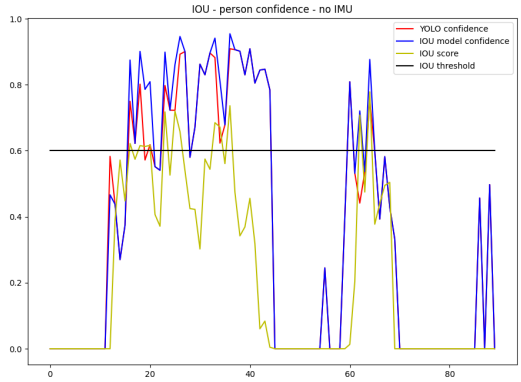

Figure 6. The IoU model without IMU recognizing the same "person" as in figure 5. All notations are the same as the previous figure. It's is notable that the yellow line here is much lower that that in figure 5, which leads to a much lower blue line.

Video	YOLO AP	IoU AP
1	1.00	0.99
2	0.85	0.84
3	0.48	0.66
4	0.49	0.45
5	0.99	0.98
6	1.00	1.00
7	0.98	0.98
8	0.98	0.97
9	0.82	0.81
10	0.94	0.95
11	0.75	0.67
12	0.81	0.57
13	0.67	0.47
14	1.00	1.00
15	1.00	1.00

Table 2. Results. Ours is better.

to stay the same or increase. This is because the IoU model produces a number of false positives when the detected object is staying still. For example, in the "person" example above, when the "person" is staying still, the IoU model will output detection results for several "person"s because of its overconfidence of the object "person" being in the video, leading to false positives.

Overall, the Intersection over Union model is extremely effective at increasing the detection confidence of true positives and producing more true positives. However, there is also a danger of producing more false positive at the same time, especially when the intended object is not moving.

6. Conclusion

This work presents two post-processing models of video object recognition using an Inertial Measurement Unit. The two models can fit any existing object recognition network and show great promise in boosting the recognition confidences. It shows that an IMU is particularly effective at a low frame rate of 1 fps, enabling the system to achieve a result that's comparable to a high frame rate, saving significant computing power and time.

Future work on this could be directed in several aspects. Firstly, a really big constraints on the two models in this paper is that it assumes the objects in the video are mostly not moving. Vision approaches such as guided flow could be used to remove this constraint and add to the generality of the models. Secondly, one could consider methods to increase the true positive detections and decrease false positives, which the two current models aren't very good at. Thirdly, the two models could be fused together for combined benefits.

References

[1] M. Alatise and G. Hancke. Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter. *Sensors*, 17(10):2164, Sep 2017.

[2] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, 16(7):1912–1919, 2007.

[3] M. Bicego, U. Castellani, and V. Murino. A hidden markov model approach for appearance-based 3d object recognition. *Pattern Recognition Letters*, 26(16):2588 – 2599, 2005.

[4] F. Dadashi, A. Arami, F. Crettenand, G. P. Millet, J. Komar, L. Seifert, and K. Aminian. A hidden markov model of the breaststroke swimming temporal phases using wearable inertial measurement units. In *2013 IEEE International Conference on Body Sensor Networks*, pages 1–6, 2013.

[5] K. H. Fielding and D. W. Ruck. Spatio-temporal pattern recognition using hidden markov models. *IEEE Transactions on Aerospace and Electronic Systems*, 31(4):1292–1300, 1995.

[6] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang. Seq-nms for video object detection. *CoRR*, abs/1602.08465, 2016.

[7] J. Hornegger, H. Niemann, D. Paulus, and G. Schlottke. Object recognition using hidden markov models. In E. S. GELSEMA and L. S. KANAL, editors, *Pattern Recognition in Practice IV*, volume 16 of *Machine Intelligence and Pattern Recognition*, pages 37 – 44. North-Holland, 1994.

[8] V. Kubala. Hidden markov model for temperal smoothing. June 2017.

[9] A. Masiero and A. Vettore. Improved feature matching for mobile devices with imu. *Sensors (Basel, Switzerland)*, 16, 2016.

[10] H. Ovrén and P.-E. Forssén. Spline error weighting for robust visual-inertial fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, June 2018. Computer Vision Foundation. VR Project: Learnable Camera Motion Models, 2014-5928.

[11] G. Panahandeh, N. Mohammadiha, A. Leijon, and P. Händel. Continuous hidden markov model for pedestrian activity classification and gait analysis. *IEEE Transactions on Instrumentation and Measurement*, 62(5):1073–1083, 2013.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2015.

[13] H. Rong, C. Peng, Y. Chen, L. Zou, Y. Zhu, and J. Lv. Adaptive-gain regulation of extended kalman filter for use in inertial and magnetic units based on hidden markov model. *IEEE Sensors Journal*, 18(7):3016–3027, 2018.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[15] W. Wan, H. Liu, L. Wang, G. Shi, and W. Li. A hybrid hmm/svm classifier for motion recognition using imu data. pages 115 – 120, 01 2008.

[16] J. Wu, L. Sun, and R. Jafari. A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. *IEEE Journal of Biomedical and Health Informatics*, 20(5):1281–1290, 2016.