Multimodality Weight and Score Fusion for SLAM

Thangarajah Akilan, *Member, IEEE*, Edna Johnson, Gaurav Taluja, Japneet Sandhu, Ritika Chadha

***Abstract*—Simultaneous Localization And Mapping (SLAM) is used to predict the trajectory by the Autonomous Navigation Robots (ANR), for instance Self-Driving Cars (SDC). It computes the trajectory through sensing the surroundings, like a visual perception of the environment. This work focuses on the performance improvements of a SLAM model using multimodal learning: (i). early fusion via layer weight enhancement of feature extractors, and (ii). late fusion via score refinement of the trajectory (pose) regressor. The comparative analysis on Apolloscape dataset shows that the proposed fusion strategies improve localization performance significantly. This work also evaluates applicability of various Deep Convolutional Neural Networks (DCNNs) for SLAM.**

***Index Terms*—SLAM, deep learning, multimodal learning**

I. Introduction

SLAM is a technology that is generally employed in autonomous vehicles to form the trajectory based on the current location coordinates and the view from that particular location [1]. The applications of SLAM range from self-driving to spatial exploration. The core components of SLAM for Self-Driving Vehicles (SDV) include the understanding of the vehicle's environment and the location of the vehicle, such as the coordinates that help in the formation of a trajectory and its interpretation of the view at an instance. Although there are many techniques to do self-localization and mapping, the Deep Learning (DL)-based approaches have surmounted others by their efficiency in extracting the finest features and giving better results comparatively.

This research aims at the improvement of a self-localization module based on PoseNet architecture on Apolloscape dataset. For our knowledge, there is no research work has been fully explored the performance analysis of various self-localization modules on the Apolloscape dataset. The proposed solution consists of two parts, wherein the first part focuses on an extensive experimental study of several DL models for SLAM. The second part is dedicated for ablation analysis of the two proposed multimodal fusion approaches: (i). early fusion via layer weight enhancement of feature extractor, and (ii). late fusion via score refinement of the trajectory regressor. The proposed models harness five pretrained DCNNs, viz., ResNet18, ResNet34, ResNet101, VGG16 and VGG19 as the feature extractor [2] of the pose regressor module.

The ablation analysis shows that the early fusion model achieves the best performance with $14.842m$ of translation error and $0.673°$ of rotation error. While, the late fusion model achieves the best performance with $9.763m$ of translation error and $0.945°$ of rotation error. Compared to the best DCNN unimodals, the proposed multimodality early and late fusion models gain 9% and 40% of improvements in translation and 19% and 2% of improvements in rotation respectively.

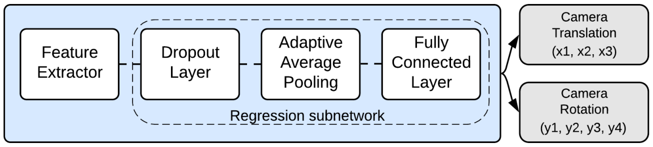


Fig. 1: PoseNet Architecture Consisting of a Feature extractor and a Pose-regressor Sub-modules.

The rest of this paper is organized as follows. Section II provides the basics of SLAM, PoseNet regressor and performance matrices used for SLAM. Section III elaborates the proposed fusion models, while the experimental setup and ablation study are given in section IV. Section V concludes the research work with the future directions.

II. Background

A. SLAM

The goal of SLAM is to estimate the trajectory of the vehicle, and building a map by passing visual inputs [3]. It has gained worldwide attention in mobile robotics for past few decades. Currently, Global Positioning System (GPS) is the most prevalent method for self-localization using satellites for location information [4]. Nonetheless, map-based studies, like OpenStreetMaps[1] has opened innumerable gates for advanced research in SLAM.

There are two main approaches for localization of an autonomous vehicle: metric SLAM and appearance-based SLAM [5]. However, this research focuses on an appearance-based SLAM that is trained in an end-to-end fashion by giving a set of visuals collected from discrete locations.

B. PoseNet

PoseNet is a DCNN, which takes an image as the input and regresses the poses of the image taken. The block diagram shown in Fig. 1 depicts the basic architecture of the PoseNet. It subsumes a feature extractor and a regression subnetwork. The feature extractor can be a pretrained DCNN, like ResNet34, VGG16, or AlexNet. The regression subnetwork consists of an average pooling, a dropout, and fully connected layers networked sequentially. It receives a high dimensional feature vector from the feature extractor. Through the average pooling and dropout layers, it is then reduced to a lower

T. Akilan, E. Johnson, G. Taluja, J. Sandhu, and R. Chadha are with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada. (e-mail: {takilan, ejohnso8, gtaluja, rchadha1, jsandhu6, }@lakeheadu.ca).

[1] https://www.openstreetmap.org/

978-1-7281-5442-8/20/$31.00 ©2020 IEEE

dimension for generalization and faster computation in the PoseNet model [6]. The poses are in 6-Degree of Freedom (DoF), which define the six parameters in translation and rotation [5]. The translation parameters include forward/backward ($x-axis$), left/right ($y-axis$), and up/down ($z-axis$). The rotation parameters include yaw ($normal-axis$), pitch ($transverse-axis$), and roll ($longitudinal-axis$). These are the core parameters that help in getting the accurate position of the vehicle and then form the trajectory.

C. Training Objective

The regressor subnetwork is trained to minimize the translation and rotation errors. The objective of this training is governed by the single loss function (L_β) as given in (1):

$$L_\beta(I) = L_x(I) + \beta L_q(I), \tag{1}$$

where L_x, L_q are the losses of translation and rotation respectively, and I is the input visual. β is a scaling factor that is used to balance both the losses [7]. β is calculated using homoscedastic uncertainty that combines the losses as defined in (2):

$$L_\sigma(I) = \frac{L_x(I)}{\hat{\sigma}_x^2} + \log \hat{\sigma}_x^2 + \frac{L_q(I)}{\hat{\sigma}_q^2} + \log \hat{\sigma}_q^2, \tag{2}$$

where $\hat{\sigma}_x$ and $\hat{\sigma}_q$ are the uncertainties for translation and rotation respectively. Here, the regularizers $\log \hat{\sigma}_x^2$ and $\log \hat{\sigma}_q^2$ prevent the values from becoming too big [7]. The loss while training the model can be calculated using a stable derivation as shown in (3):

$$L_\sigma(I) = L_x(I)^{-\hat{s}_x} + \hat{s}_x + L_q(I)^{-\hat{s}_q} + \hat{s}_q, \tag{3}$$

where the learning parameter $s = \log \hat{\sigma}^2$. The values $\hat{s_x}$ and $\hat{s_q}$ are set to $\hat{s_x} = 0$ and $\hat{s_q} = -3.0$ to get the best results according to [7].

D. Multimodality fusion

Multimodality fusion is one of the effective and widely adopted techniques used across various learning systems to improve the performance [8]–[10]. Whereby, different models or the complementary cues from various models are combined using strategies, such as mathematical modeling and subspace transformation [11].

III. PROPOSED METHOD

A. Preprocessing

Before passing the images and poses to the PoseNet model, it is required to preprocess the data adequately. The preprocessing involves checking the consistency of the images, resizing and center cropping of the images, extraction of mean and standard deviation, and normalization of the poses. The images are resized to 260 × 260 and center cropped to 250 × 250. From the ground truth of translation values, the minimum, maximum, mean, and standard deviation are computed. The rotation values are read as Euler angles which suffer from wrap around infinities, gimbal lock and interpolation problems. To overcome these challenges, Euler rotations are converted to seven quaternions [12].

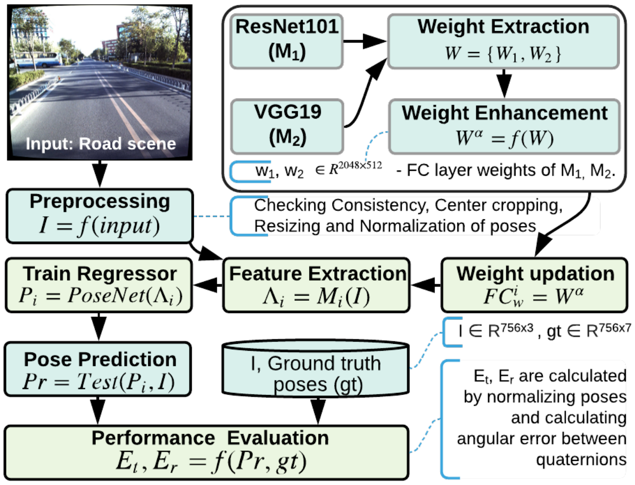

Input: Road scene
ResNet101 (M$_1$)
VGG19 (M$_2$)
Weight Extraction $W = \{W_1, W_2\}$
Weight Enhancement $W^\alpha = f(W)$
w$_1$, w$_2$ $\in R^{2048\times512}$ - FC layer weights of M$_1$, M$_2$.
Preprocessing $I = f(input)$
Checking Consistency, Center cropping, Resizing and Normalization of poses
Train Regressor $P_i = PoseNet(\Lambda_i)$
Feature Extraction $\Lambda_i = M_i(I)$
Weight updation $FC_w^i = W^\alpha$
I $\in$ R^{756x3}, gt $\in$ R^{756x7}
Pose Prediction $Pr = Test(P_i, I)$
I, Ground truth poses (gt)
E$_t$, E$_r$ are calculated by normalizing poses and calculating angular error between quaternions
Performance Evaluation $E_t, E_r = f(Pr, gt)$

Fig. 2: Early Fusion via Feature Extractor Weight Enhancement.

B. Proposed Fusion Models

The proposed solution introduces two strategies to enhance the performance: (i). early fusion via layer weight enhancement of the feature extractors and (ii). late fusion via score refinement of the pose regressor.

1) Early fusion via layer weight enhancement (E_{fusion}): The operational flow of the early fusion via layer weight enhancement during training is shown in Fig. 2. ResNet101 and VGG19 are selected as feature extractors. These two models are selected based on their individual performances on the Apolloscape test dataset. They have recorded the top-2 best results as tabulated in Table I. The weights from the Fully Connected (FC) layer are extracted from each model, viz. VGG19 and ResNet101 as shown in the equations (4) and (5) respectively.

$$FC_{V \rightarrow VGG19} = f\left(\sum_{i=1}^{N} w_i^V x_i + b^V\right), \tag{4}$$

where $f(\cdot)$, N, i, w_i^V, x_i, and b^V are activation function, total number of neurons in the FC_V layer, neuron index, weight of i^{th} neuron of FC_V layer, input to the FC_V Layer, and bias of VGG19's FC layer respectively.

$$FC_{R \rightarrow ResNet101} = f\left(\sum_{i=1}^{N} w_i^R x_i + b^R\right), \tag{5}$$

where $f(\cdot)$, N, i, w_i^R, x_i, and b^R are activation function, total number of neurons in the FC_R layer, neuron index, weight of i^{th} neuron of FC_R layer, input to the FC_R layer, and bias of ResNet101's FC layer respectively. The FC layer weights of the two feature extractors are fused by using arithmetic operations as defined in the equation (6).

$$\hat{W} = f(W^V, W^R), \tag{6}$$

where $f(.)$, W^V, W^R are operations like arithmetic addition or multiplication, weight of the FC layer of VGG19 and weight of the FC layer of ResNet101 respectively. The fused values are used to update the weights of FC layer of the ResNet101 feature extractor as given in equation (7).

$$FC_m = f\left(\sum_{i=1}^{N} \hat{W}_i x_i + b^m\right), \tag{7}$$

where FC_m, N, $\hat{W}_i$, x_i, and b^m denote the updated dense layer, total number of neurons in the updated dense layer, weight of i^{th} neuron of updated model, input of dense layer and bias for the updated model respectively. The updated model is used as new feature extractor for the regressor subnetwork. Then, the regressor is trained on the newly extracted features using the weight updated model.

2) Late fusion model (L_{fusion}): Figure 3 elaborates the multimodality late fusion via score refinement approach. The camera feed and the poses are preprocessed as described in subsection (III-A). Then, the pretrained CNN models are used to extract the features from the preprocessed data. In this work, the following five pretrained CNN models ResNet18, ResNet34, ResNet101, VGG16, and VGG19 are used to extract the features. Using the individual feature vectors from the aforesaid pretrained CNNs, five different PoseNet models are trained independently. Then, pose predictions are carried out on the test samples using trained PoseNet models, $M_{i=\{1,\cdots,5\}}$ simultaneously as shown in (8).

$$Pr_i = M_i(I), \tag{8}$$

where Pr_i, M_i, and I denote the predicted poses, the i^{th} trained PoseNet model, and the test input visual respectively. The predicted values in (8) are fused to achieve the refined poses of translation and rotation as

$$Pr = f\left(Pr_i\right), \tag{9}$$

where Pr is the distilled poses after late fusion using the average arithmetic operation $f(.)$.

IV. EXPERIMENTAL SETUP AND ANALYSIS

A. Dataset

Apolloscape self-localization dataset is used for many computer vision tasks related to autonomous driving [13]. This dataset consists of 3000 stereo vision road scenes captured from a moving vehicle. For each image there is a ground truth of poses with 6-DoF. From this entire dataset, a mutually exclusive training and test sets are created with the ratio of 3 : 1.

B. Training Configurations

The hyper-parameter values have been fixed while performing the experimental tasks on different models to avoid ambiguities in the comparative study. The learning rate used in all the models is 0.01 with the drop out rate 0.5. The batch size of the images is 34 and every model is trained for 1000 epochs with the optimizer as Adam to ensure the uniform comparison across different models.

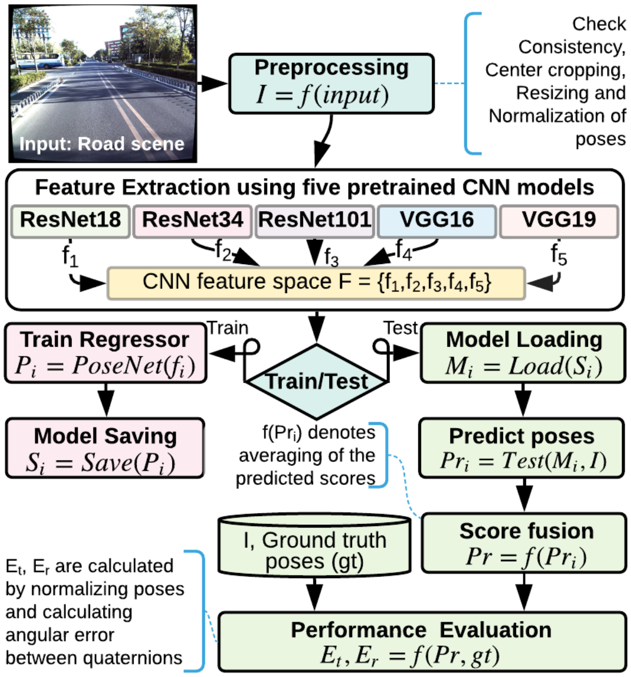

Input: Road scene
Preprocessing
$I = f(input)$
Check Consistency, Center cropping, Resizing and Normalization of poses
Feature Extraction using five pretrained CNN models
ResNet18
ResNet34
ResNet101
VGG16
VGG19
f_1
f_2
f_3
f_4
f_5
CNN feature space F = {f_1,f_2,f_3,f_4,f_5}
Train Regressor
$P_i = PoseNet(f_i)$
Train
Train/Test
Test
Model Loading
$M_i = Load(S_i)$
Model Saving
$S_i = Save(P_i)$
f(Pr_i) denotes averaging of the predicted scores
Predict poses
$Pr_i = Test(M_i, I)$
I, Ground truth poses (gt)
Score fusion
$Pr = f(Pr_i)$
E_t, E_r are calculated by normalizing poses and calculating angular error between quaternions
Performance Evaluation
$E_t, E_r = f(Pr, gt)$

Fig. 3: Late Fusion Model via Predicated Pose Refinement.

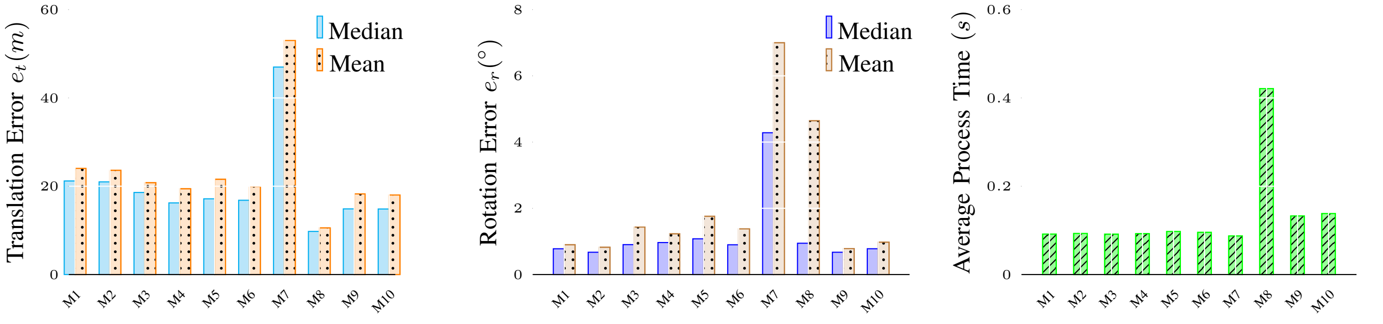
Models	Median $e_t(m)$	Mean $e_t(m)$	Median $e_r(°)$	Mean $e_r(°)$	MAPT (s)
M1 - ResNet18	21.194	24.029	0.778	0.900	0.091
M2 - ResNet34	20.990	23.597	0.673	0.824	0.093
M3 - ResNet50	18.583	20.803	0.903	1.434	0.091
M4 - ResNet101	16.227	19.427	0.966	1.230	0.092
M5 - VGG16	17.150	21.571	1.079	1.758	0.097
M6 - VGG19	16.820	19.935	0.899	1.378	0.095
M7 - AlexNet	46.992	53.004	4.282	7.177	0.087
M8 - L_f	9.763	10.561	0.945	4.645	0.421
M9 - E_f^+	14.870	18.256	0.673	0.784	0.132
M10 - E_f^*	14.842	18.013	0.779	0.977	0.138

TABLE I: Performance analysis. Proposed models: M8, M9 and M10. MAPT - mean average processing time per sample.

C. Performance Analysis

The Apolloscape dataset is run through different DCNNs, including ResNet18, ResNet34, ResNet50, ResNet101, VGG16, VGG19, and AlexNet.

1) Translation and Rotation Errors: The performance of these models with PoseNet are compared with the proposed multimodality fusions as given in Table I, where M8, M9, and M10 stand for the proposed late fusion (L_f), early fusion with addition (E_f^+), and early fusion with multiplication (E_f^*) respectively. The results obtained are the mean and median of translation errors and rotation errors. The translation error is measured in meters (m), while the rotation error is measured in degrees ($°$). As observed from the results, the late fusion model shows better performance than the unimodal, but not



(a) Error in terms of Translation.

(b) Error in terms of Rotation.

(c) Average Processing Time per Sample.

Fig. 4: Performance Analysis Across All the Models.

better when compared with the early fusion models. The late fusion shows a 52% decrease in the translation's median error and a 70% decrease in the translation's mean error when compared to the early fusion using addition that has ResNet101 as a feature extractor.

The median of rotation errors in late fusion model shows a 29% increase and the mean of rotation error shows an 83% increase when compared with the early fusion model using addition. Although the early fusion using addition gives similar mean and median of translation error as early fusion using multiplication, the former gives 17% lesser median of rotation error and 24% lesser mean of rotation error. Therefore, early fusion with addition (E_f^+) achieves better results when compared to all the models. Figure 4(a) and 4(b) compare mean and median errors of translation and rotation of all the models.

2) Timing Analysis: The timing analysis is carried out on a machine with an Intel Core i5 processor that uses the Google Colaboratory cloud programming environment having a Tesla K80 GPU with 2496 CUDA, a hard disk space of 319GB and 12.6GB RAM. Table I shows the mean average processing time (I_t) which is calculated by dividing the inference time with batch size of 10. As seen from the Table I and Figure 4(c), the fusion models take slightly extra time compared to the unimodality-based pose predictions. In early fusion, the weights of the models under study are fused together before actually training it, thereby making it to have an overhead of processing time.

V. Conclusion

This research proposes two multimodal fusion strategies to improve the localization accuracy of a SLAM model. The proposed models show excellent improvements over unimodal-based localization with a little overhead of processing time. The future work is dedicated to overcome the additional processing time used for the fusion operations through network optimization, like layer pruning and bit-quantization.

Acknowledgment

The authors thank Google for their generosity of freely providing the HPC machine learning platform, Colaboratory for research and academic purpose.

References

[1] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[2] Y. Yang, J. Q. M. Wu, X. Feng, and A. Thangarajah, "Recomputation of dense layers for the performance improvement of dcnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[3] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[4] A. G. Marcus A. Brubaker and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE PAMI 2015*.

[5] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, pp. 2938–2946, 2015.

[6] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," pp. 627–637, 2017.

[7] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983, 2017.

[8] T. Akilan, Q. M. J. Wu, A. Safaei, and W. Jiang, "A late fusion approach for harnessing multi-cnn model high-level features," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 566–571, Oct 2017.

[9] T. Akilan, Q. M. J. Wu, Y. Yang, and A. Safaei, "Fusion of transfer learning features and its application in image classification," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–5, April 2017.

[10] T. Akilan, Q. J. Wu, and H. Zhang, "Effect of fusing features from multiple dcnn architectures in image classification," *IET Image Processing*, vol. 12, no. 7, pp. 1102–1110, 2018.

[11] J. Xu, Y. Zhao, J. Jiang, Y. Dou, Z. Liu, and K. Chen, "Fusion model based on convolutional neural networks with two features for acoustic scene classification," *Detection and Classification of Acoustic Scenes and Events 2017*, Nov 2017.

[12] P. Bouthellier, "Rotations and orientations in r3," *27th International Conference on Technology in Collegiate Mathematics*, vol. 27, 2015.

[13] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, 2019.