Meta-Learning, Fast Adaptation, and Latent Representation for Head Pose Estimation

Manoj Joshi
Institute of Engineering, Nepal
075mscsk009.manoj@pcampus.edu.np

*Dibakar Raj Pant
Corresponding Author
Institute of Engineering, Nepal
drpant@ioe.edu.np

Rupesh Raj Karn
Khalifa University, Abu Dhabi, UAE
rupesh.karn@ku.ac.ae

Jukka Heikkonen
University of Turku, Turku, Finland
jukhei@utu.fi

Rajeev Kanth
Savonia University of Applied Sciences, Kuopio, Finland
rajeev.kanth@savonia.fi

Abstract—**Head pose estimation is used in a variety of human-computer interface applications, like stare tracking, driving assistance, impaired assistance, and entertainment. Advances in convolutional neural networks have a considerable improvement in the performance of head pose estimation. However, difficulties in capturing well-labelled head pose data and differences in the facial features of different persons make them difficult to use. This work proposes a meta-learning based technique for head pose estimation problem in BIWI head pose dataset. An approach to learning latent representation of head pose features using variational autoencoder is implemented. Then a fast, adaptable head pose estimator is trained using meta-learning in a few-shot settings. Model agnostic meta-learning (MAML) algorithm has been deployed for training a head pose estimator. Mean Average Error (MAE_{avg}) of 7.33 is achieved in predicting head pose angles in one-shot settings. After meta-training, the optimized model is used to analyze fast adaptation in a test set that has been separated from the BIWI head pose dataset. We begin with the trained network's optimum parameters and optimize the inner loop for quick adaptation. The optimized model can predict accurate head poses using as few as 10 gradient descent steps in the unseen set of tasks sampled from the test set.**

I. Introduction

In recent years, a lot of progress has been made in the field of internet of things, robotics, image processing, augmented reality, and human-machine interaction. The advancement in gaze tracking, driving assistance and impaired assistance requires a robust head pose estimation system. A well-built head pose estimation is generally used for understanding human attention [1], their social interactions, and behavior [2] and has been widely researched and explored in the cognitive psychology and neurophysiology communities. A driver assistance system might use head pose estimate to decelerate the vehicle while pedestrians are unaware of the vehicle's presence in self-driving vehicles [3]. The need of good head pose estimators is not limited to these domains. Significant applications have been made in surveillance and anomaly detection, human-computer interaction, and crowd behavioral dynamics study [4]. Head pose estimation is a difficult problem to solve in "in-the-wild" settings, such as extreme orientations, illumination variation, varying resolution, and the presence of hairs on the face and makeup.

Traditional image processing based methods acquired some success in estimating head pose. Methods such as Histogram of Oriented Gradients (HOG) [5] successfully estimated the head orientation from images and videos. Most of these methods are based on discriminative/landmark-based or parameterized appearance based models. Despite being good estimators of head pose's angles, novel machine learning approaches have been proposed because of their flexibility and robustness to extreme head pose changes.

Convolutional Neural Networks (CNN) have been an efficient and most popular choice to develop robust head pose estimators [6]–[8]. The high efficiency of CNN's is highly reliant on training a network with a large number of well-annotated instances of head pose data with a variety of visual variations. A well-annotated large head pose dataset is challenging to obtain in many cases. Also, a good head pose estimator should achieve the same proficiency as CNN's by rapidly understanding and adapting from fewer examples and continuing to adapt as more evidence becomes available.

Meta-learning is an alternative paradigm in which a learning model acquires experience across numerous learning episodes from a variety of related tasks. Then it uses the knowledge gained to improve its learning performance in future. Meta-learning algorithms are supposed to address these challenges using a few-shot learning settings [9]. Meta-learners learn a new task from limited amount of data. This *'learning-to-learn'* can result in a range of benefits, including increased data and compute efficiency. Meta-learning is more similar to human and animal learning [10], where learning techniques improve with time.

In this article we present a meta-learning-based approach for head pose estimation. We propose a framework to estimate head poses using very few calibration samples. It consists of: i) Face detection from head pose images using multi-task cascacaded convolutional neural network (MTCNN) [11] ii) learning a latent representation of human faces using a variational autoencoder [12] iii) use latent features to a train a

ISSN 2305-7254

meta-learner using a model-agnostic meta-learning algorithm (MAML) [13] iv) adapt to new faces with good performance using very few samples. Our head pose estimation framework was evaluated using the *in-the-wild* BIWI head pose dataset [14]. The proposed framework successfully adapted to new faces using a few samples ($k \leq 9$) to produce accurate head pose estimates.

II. Related Works

Some early methods for estimating head pose are based on appearance template methods [15], [16], which used image-based comparison metrics. Detector arrays-based method was developed for frontal face detection [17]. Instead of directly comparing images to templates, it used a detector trained on many images using supervised learning algorithms to detect head pose. Geometric models [18] used facial key-points to compute the head pose. The primary constraint of this technique is landmark detection [19]–[22]. Nonlinear regression methods were developed to detect head poses by learning a nonlinear function that can map an image space to one or more head pose directions [23]. High dimensional image data were handled by principal component analysis (PCA) such as in [24], [25], whereas neural networks [26] were used for learning nonlinear functions. Classification-based methods were also developed to estimate head pose using discretized sets of head poses. Such methods used random forest algorithms [27], [28], multi-task learning [29], and neural networks [26] to classify head poses.

Osadchy *et al.* [30] proposed a real-time CNN-based approach for head pose estimation. Their CNN architecture is similar to LeNet-5 [31] but has more feature maps. In 2014, Ahn *et al.* [32] developed a network using four convolution layers and two fully connected layers for head pose estimation in the BIWI head pose dataset. Other proposed methods use RGB images along with the depth information as seen in literature [33], where GoogleLeNet [34] was used to train the model. Venturelli *et al.* [35] proposed a shallow network with five convolutional layers and three fully connected layer with improved performance. Ruiz *et al.* [26] used ResNet50 architecture with three mean squared error (MSE) and cross-entropy loss for each head pose angles as evaluation metrics. Recent work on head pose estimation has been proposed by Patacchiola [36] on Prima and AFLW datasets.

The above-stated literature with CNN-based methods gives excellent head pose estimation results but requires a lot of training samples to find the best estimator, lack generalization in the unknown task, and have poor adaptation to a new set of task. Finn *et al.* [13] have proposed a model-independent algorithm for meta-training. Any model that utilizes gradient descent is compatible with model-agnostic meta-learning. Sun *et al.* [9] have proposed meta-learning as a framework that can perform well on few-shot learning setup. The basic idea is to learn how to adapt a base-learner to a new task with only a few labeled samples by using a large number of identical few-shot tasks. Antoniou *et al.* [37] have introduced methods to train a meta-learning algorithm such as model agnostic meta-learning (MAML) for adapting to tasks such as regression and classification. Park *et al.* [38] have proposed a novel architecture for few-shot gaze estimation using meta-learning. They have used very few calibration samples for training a meta-learner for person-specific gaze estimation. Their method also can adapt to any new person with much more accuracy in gaze estimation. Most of the research for head pose estimation is based on CNN-based learning models. This work uses a meta-learning based method that learn how to learn accurate head pose using very few examples. In the framework of meta-learning, we cast head pose estimation as a multi-task problem, with each subject treated as a new task for the meta-learner.

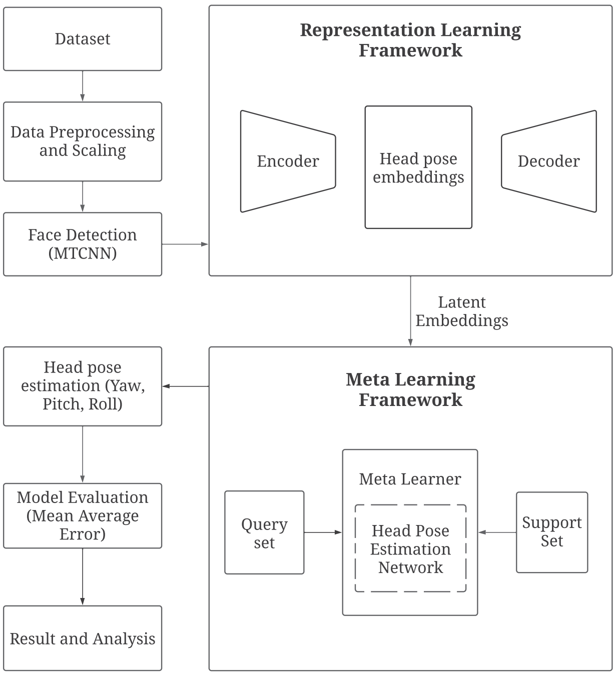
Dataset
Data Preprocessing and Scaling
Face Detection (MTCNN)
Representation Learning Framework
Encoder
Head pose embeddings
Decoder
Latent Embeddings
Head pose estimation (Yaw, Pitch, Roll)
Meta Learning Framework
Model Evaluation (Mean Average Error)
Meta Learner
Query set
Head Pose Estimation Network
Support Set
Result and Analysis

Fig. 1. Illustration of head pose estimation architecture using meta-learning. The cropped faces are sent to representation learning framework where a variational autoencoder generates latent embeddings. The embeddings are then passed into a meta-learning framework which uses a head pose estimator network to estiamte the head pose angles.

III. Method

An MTCNN network is used to detect and crop the faces from the BIWI head pose dataset. The cropped faces are then sent to a representation learning network comprising a simple variational autoencoder (VAE) network. The variational autoencoder learns the important features from training images and generates 200 latent features of each training sample. Generating latent representations of head poses allows us to design a simpler network for meta-training. A simple deep neural network is designed to train a meta-learner using Model-Agnostic Meta-Learning (MAML) [13] algorithm.

The datasets used, preprocessing of data, the architecture of proposed framework, and model evaluation strategy are briefly discussed in the following subsections.

A. Dataset

Well-known BIWI head pose [14] benchmark dataset is used for training a meta-learner using MAML for head pose estimation. The BIWI head pose dataset comprises of 15,678 images of 20 subjects. A depth image that corresponds to an RGB image with a size of 640 × 480 pixels is provided, along with annotations for head pose angles. The head pose change ranges approximately ±75° in yaw, ±60° in pitch and ±50° in roll. Each image's ground truth is given in the form of the 3D position and rotation of the head. Fig 2 shows samples from BIWI head pose dataset.

Fig. 2. Example of images from Biwi head pose dataset [39]

Because of the large size of RGB images and unnecessary background objects, the faces of subjects were detected and cropped using MTCNN [11] model and stored separately for training and testing the meta-learner. 10581 cropped images from fifteen subjects were used to train a meta-learner, and 2638 images from the remaining five subjects were used to test the model performance. Images with very occluded backgrounds, multiple objects and extreme poses were removed when detecting the faces using MTCNN [11].

B. Experimental Testbench

All the experiments in this article are performed using Python 3.8.0. We used PyTorch and PyTorch Lightening [40] for experimenting with the architecture. Ray Tune [41] library is used for hyperparameter tuning and neural-architectural-search (NAS) [42]. The Google Colab platform is used to train the variational autoencoder and meta-learning model. OpenCV [43] library is used extensively for image processing tasks in the experiments.

C. Proposed Architecture

In this article, we propose a three-stage architecture for head pose estimation *i.e. Face detection, Representation learning and Meta-learning*. Fig 1 shows the proposed head pose estimation network architecture using meta-learning. Each stages in head pose detection are discussed below.

1) Data pre-processing:

a) Face Detection: It is challenging to detect face and face alignment from images in varying unconstrained environment. Varying lighting conditions, visual variations in human faces and extreme head pose variations are the major challenges to detect face properly from the images. It requires face detection, localization and computation of bounding box coordinates to get the exact coordinates of face. Face detection is one of the most essential process in the proposed architecture hence, we used a multi-task cascaded convolutional neural network (MTCNN) [11] to detect faces and get their bounding box coordinates.

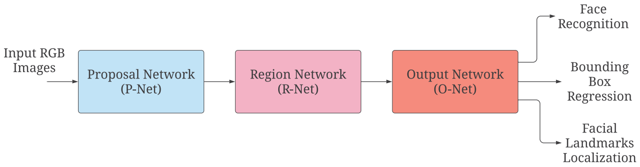

Fig. 3. Architecture of MTCNN for face detection

Fig 3 shows the architecture of the MTCNN [11] network. RGB images are sent to three stages in MTCNN, i.e., P-Net, R-Net and O-Net. The O-Net layers outputs face classification, bounding box regression and facial landmarks. Using MTCNN [11], we get cropped faces of size 128×128 which we use to train our representation learning framework.

b) Data Normalization: Cropped images were normalized to get all pixel values in the range of 0 and 1. The cropped images were scaled to a new range using a min-max scaler before feeding into the representation learning framework.

2) Representation Learning Framework: Representation learning seeks to obtain a usable representation of data. This is also known as feature learning since it is capable of learning relevant characteristics of the data. In this work, representation learning was utilized after the face detection process to get the latent features from cropped images. The representation learning framework should be able to preserve the facial and head pose features of the training samples. For this, we can use any representation learning methods, but we focus on implementing variational autoencoders (VAE) since VAE's link representation learning to generative modeling, i.e., they make it possible to create valuable data from scratch. Fig 5 shows the basic architecture of representation learning framework using a variational autoencoder.

For each input image x, an encoder is defined as $E : x \rightarrow z$ and a decoder $D : z \rightarrow \hat{x}$ such that $D(E(x)) = \hat{x}$. In this work, a variational autoencoder model efficiently learns the important features from training images and provides 200 latent features. The encoder comprises of five convolution

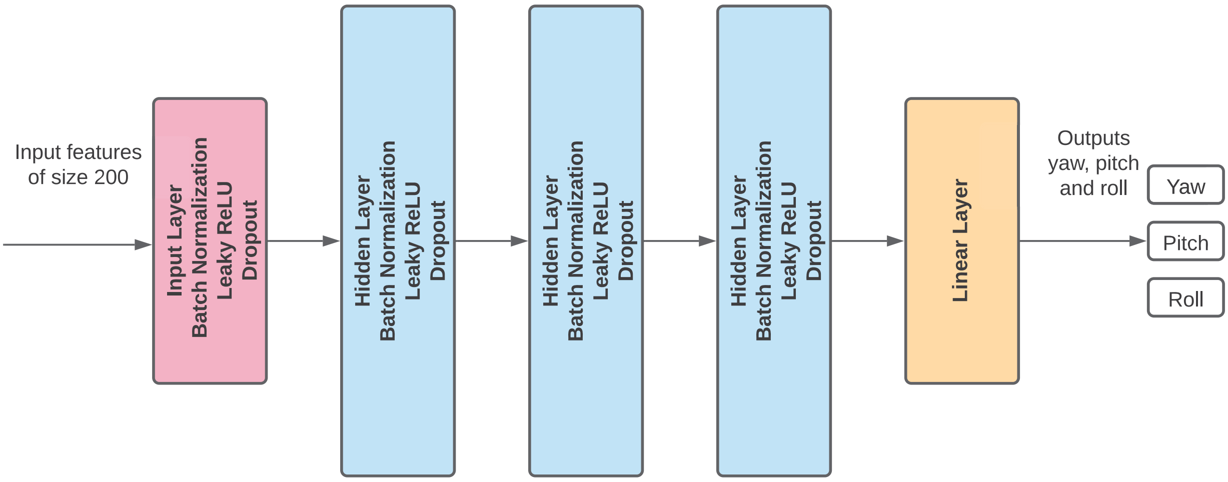

Fig. 4. Basic architecture of meta learner network for head pose estimation

layers with LeakyReLu as an activation function. The dropout rate of 0.25 and a stride of size two has been considered to construct an encoder. The kernel size of three has been used for all the convolution layers except the last one in which kernel size of one is considered. The results of convolutional layers are flattened by a Flatten layer. The encoder results in 200 latent embeddings. The decoder comprises a linear layer that receives 200 features and does a linear transformation to get 4096 features. Four ConvTranspose2D layers are added having LeakyReLu as an activation function. The ConvTranspose2D layers have a stride of two, kernel size of three, and a dropout rate of 0.25. The last ConvTranspose2D layer reconstructs output images of size 128 × 128. After training 50 epochs, we got the latent representations z of size 200, which will be input to our meta-learner. The number of filters, kernel size, and dropout rates of convolution layers were considered as hyperparameters for the variational autoencoder which were optimized during hyperparameter tuning.

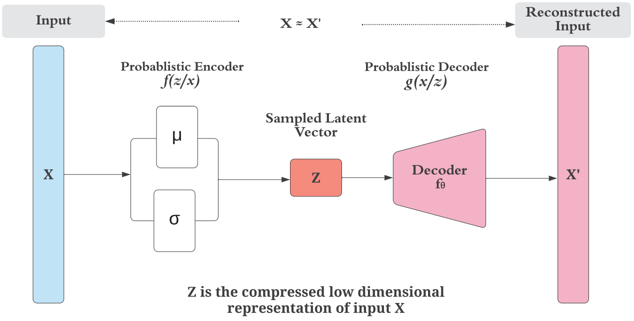

Fig. 5. The basic architecture of variational autoencoder (VAE)

3) Meta-Learning Framework: Model-agnostic meta-learning (MAML) [13] is a model-agnostic and task-agnostic algorithm capable of training model parameters quickly for fast adaptation to new tasks. The beauty of this algorithm is that it learns new tasks quickly by using very few gradient updates in the model.

Algorithm 1 Model-Agnostic Meta-Learning for Head pose estimation

```
Require: $D(\tau)$ : distribution over tasks (human faces)
Require: $\alpha, \beta$ : learning rates
Randomly initialize parameter $\omega$
while not done do
    Sample batch of task $T_i \sim D(\tau)$
    for all $T_i$ do
        Evaluate $\nabla_{\omega} L_{T_i}(f_{\omega})$ with respect to K samples
        Calculate adapted parameters using gradient descent: $\omega^{'} = \omega - \alpha \nabla_{\omega} L_{T_i}(f_{\omega})$
    end for
    Update $\omega \leftarrow \omega - \beta \nabla_{\omega} \Sigma_{T_i \sim D(\tau)} L_{T_i}(f_{\omega'})$
end while
```

As observed in Algorithm 1, a model f_{ω} is considered with parameters ω having τ number of tasks. The model f_{ω} is trained using tasks T_i taken from distribution $D(\tau)$, taking only K samples at a time. This training resulted in a robust few-shot learner, which can now be used to generalize new samples taken from the entirely new task set T_i. The model f_{ω} is regularly updated to minimize loss L_{T_i} for task set T_i.

The proposed head pose estimator uses a model trained using MAML [13] algorithm. We consider *M* as a head pose estimation model and then after training, this estimator learns the optimal weights ω^{*}. Once the model *M* learns ω^{*} it is said to be optimized. In other words, the model M_{ω}^{*} is fine-tuned with very few examples of a new person *P* which was never seen during model training and can generalize to validation examples of same person. For learning optimal weights ω^{*}, we need to setup a few-shot learning setup. For this we created meta-training (D^{train}) and meta-testing (D^{test}) subset of non-overlapping subjects from the entire set of subjects *D*. In each meta training step *t*, a person p^{train} is selected from D^{train}. A meta training sample by random sampling is created for the selected person defined as $p^{train} = \{S_s^{train}, S_q^{train}\}$. The subset $S_s^{train} = \{(z_i, gt_i)\}$ where *i* ranges from 1 to *k* training examples is called the support set and subset $S_q^{train} = \{(z_j, gt_j)\}$ where *j* ranges from 1 to *m* examples of the same person is called query set. The latent representation of head poses *z*, and ground truth of head pose angles *gt* were used in meta training. The parameters *k* and *m* are generally chosen small ($\leq$ 20) in few shot settings. The mean absolute error (MAE) in predicting yaw, pitch, and roll, is used as a cost function during gradient update.

Meta-learning starts by computing loss for support set S_s^{train} and updating weights ω_t at step *t* using few gradient steps and learning rate α as shown below.

$$\omega_t^{'} = f(\omega_t) = \omega_t - \alpha \nabla Loss^{s}_{p^{train}}(\omega_t) \tag{1}$$

The mean absolute error (MAE) in computing head poses angles is given by:

$$Loss = \frac{1}{n} \sum_{i=1}^{n} |gt_i - \hat{y_i}| \tag{2}$$

where n is the number of samples in the support set S_s^{train}, gt_i are the ground truth of head poses angles, and $\hat{y_i}$ are the predicted head poses angles.

Using these updated weights $\omega_t^{'}$, we now compute loss for validation set S_q^{train} at step *t*. The gradients are computed with respect to original weights ω_t and using a learning rate β. Finally, the weights ω_t are updated to minimize the validation loss, as shown below.

$$\omega_{t+1} = \omega_t - \beta \nabla Loss^{q}_{p^{train}}(f(\omega_t)) \tag{3}$$

The algorithm continues until the weights are converged to optimal weights ω^*.

4) Fast Adaptation: Once we learn ω^*, our model is also optimized to M^*_ω and can be used to adapt on unseen examples. Sample a person p^{test} from test set S^{test}. We fine tune our model M^*_ω using *k* sample images from S_s^{test} to adapt to new examples faster as shown below.

$$\omega_{p^{test}} = \omega^* - \alpha \nabla Loss^{s}_{p^{test}}(\omega^*) \tag{4}$$

Finally we test performance of fast adaptation using sample images in validation set S_q^{test}.

The 200 latent features were passed through a linear layer that outputs 1000 features. The result of the linear layer was then passed through three densely connected layers, each resulting in 1000 features. The output from the third hidden layer was sent to fully connected linear layer that outputs the three head poses angles. LeakyReLU with a negative slope of 0.1 was taken as activation function in the model. The dropout rate of 0.25 was used during meta-training. Fig 4 shows the proposed architecture used for meta training. This regression model predicts the yaw, pitch, and roll angles.

D. Hyperparameter Tuning

The hyperparameters for meta-learner and variational autoencoder were tuned using Ray Tune [41], particularly grid search based approach. The meta-learning hyperparameters α and β were chosen 0.01 respectively, and the number of inner gradient steps was selected as ten. A dropout rate of 0.25 was selected while training the meta-learner. The learning rate and the dropout rate for the variational autoencoder were selected as 0.001 and 0.25, respectively. The number of filters, kernel size, and dropout rates of convolution layers has been searched using grid search based approach as a part of neural-architectural-search (NAS) [42]. Stochastic Gradient Descent (SGD) was used as an optimizer to train the network.

E. Model Evaluation

1) Mean Squared Loss: Mean squared error (MSE) is a metric to compute the average squared differences between original and predicted values. Mean squared error is computed as,

$$MSE = \frac{1}{n} \sum_{i=1}^{n} ||S_i - \hat{S_i}||^2 \tag{5}$$

where S_i is the original input images, and $\hat{S_i}$ is the reconstructed output image. The pixel-wise mean squared error is computed during representation learning using a variational autoencoder to produce latent embeddings.

2) Mean Absolute Error: Mean Absolute Error (MAE) is a well-known performance metric used to compute the similarity between two sets. It computes differences between ground truth head pose angles and predicted head pose angles. MAE is defined as,

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{6}$$

where y_i are the ground truth head pose angles *yaw, pitch and roll* and $\hat{y_i}$ is the predicted head pose angles.

3) Mean Average Error: The average of three mean absolute errors in predicting Euler's angles is taken as the final score to evaluate the proposed model. It is used to assess the overall performance of the proposed architecture in predicting Euler's angles *(yaw, pitch, and roll)* for head poses. Mean average error is computed as:

$$MAE_{avg} = \frac{yaw_{mae} + pitch_{mae} + roll_{mae}}{3} \tag{7}$$

where yaw_{mae}, $pitch_{mae}$ and $roll_{mae}$ are the mean absolute errors in predicting *yaw, pitch and roll* angles respectively.

F. Result and Analysis

1) Representation Learning: Input images of size 128×128 are trained using a representation learning pipeline to generate latent embeddings. When training a variational autoencoder for 200 epochs, the latent embeddings of size 200 were created for input images. Pixelwise mean squared loss is computed

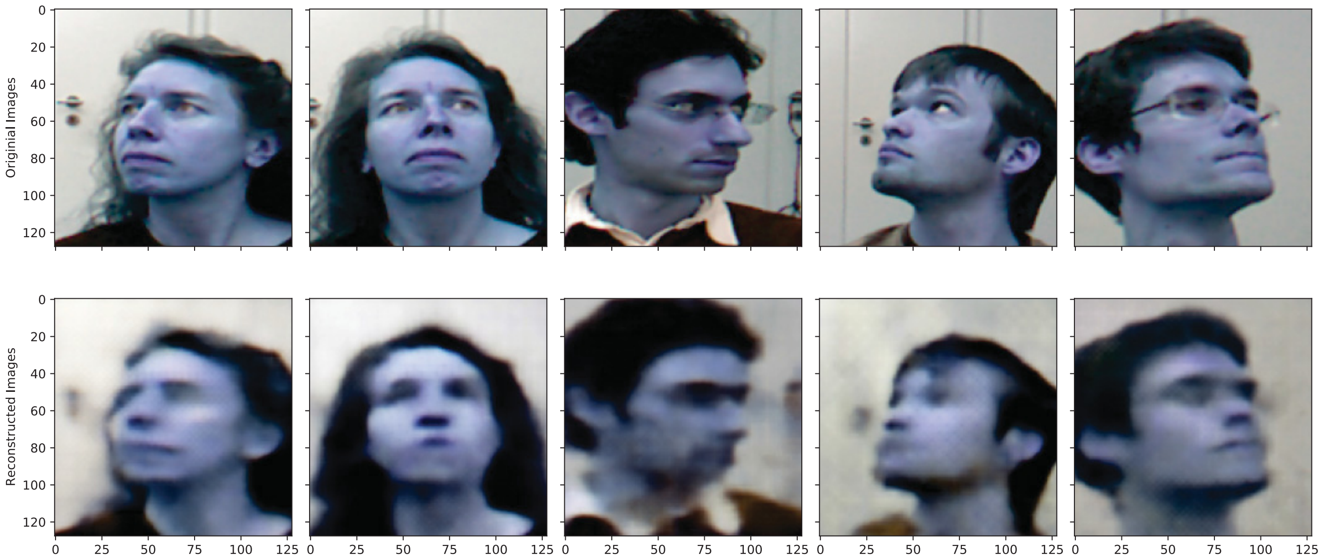

Fig. 6. Original and reconstructed images using variational autoencoder during representation learning in BIWI head pose dataset

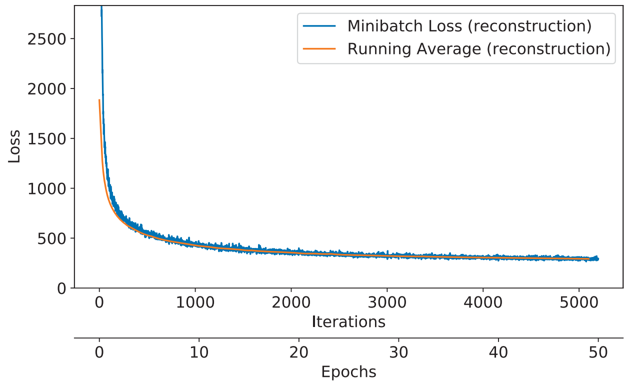

Fig. 7. Pixel wise reconstruction loss using a variational autoencoder (VAE) in terms of mean squared error(MSE)

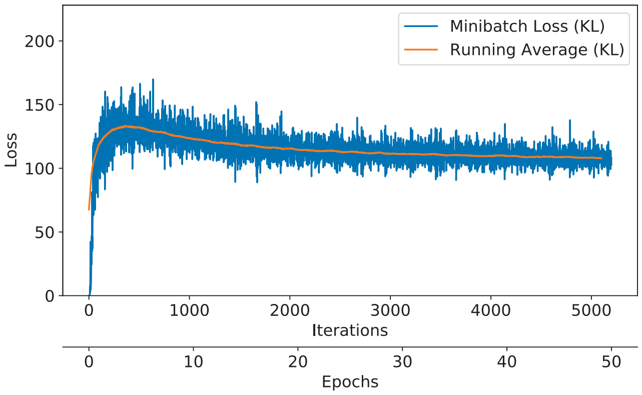

Fig. 8. KL Diveregence Loss during representation learning

to evaluate the performance of variational autoencoder for representation learning.

The reconstructed images of BIWI head pose dataset using representation learning are shown in Fig 6. Fig 7 shows the loss during regenerating original images from latent embeddings by a variational autoencoder. We can see that the mean squared error (MSE) in regenerating the pixels is decreasing with increasing epochs. Fig 8 shows the loss of Kullback–Leibler divergence (KL-Divergence) during representation learning. While training a VAE, reconstruction loss is initially given more privilege, making the KL divergence loss small. Hence, the KL-Divergence loss starts with a low value initially and increases gradually with the model training [44]. After a few epochs of training, the reconstruction loss decreases and KL-divergence loss increases, thus balancing the total loss in a variational autoencoder. The total loss of the variational autoencoder is combined reconstruction and KL-divergence loss, as shown in Fig 9. The representation learning step produced latent features of size 200, which were then used to train a meta learner.

2) One-shot settings: One sample from each of five different subjects are selected at random to create a support and query set to train MAML in one-shot settings. The support set is used for training, and the query set is used to fine-tune the network parameters. Model performance has been evaluated using Mean Absolute Error (MAE) to predict head pose angles. After training the meta learner using the MAML algorithm for 250 episodes with the meta batch size of 64, we successfully predicted the Euler angles for head poses with a mean average error (MAE_{avg}) of 7.33. Fig 10 shows the meta training and validation loss in terms of MAE.

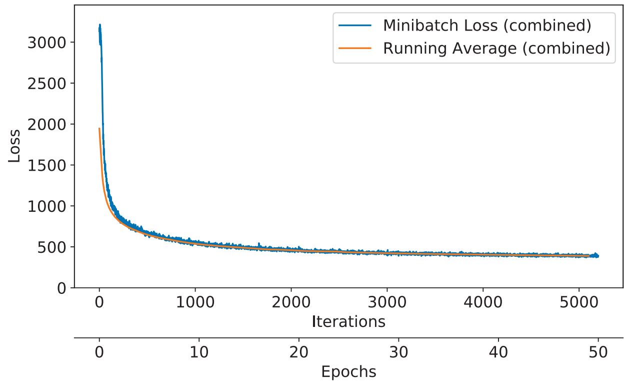

Fig. 9. Total Loss during representation learning. Total loss decreases and stabilizes along with training the variational autoencoder.

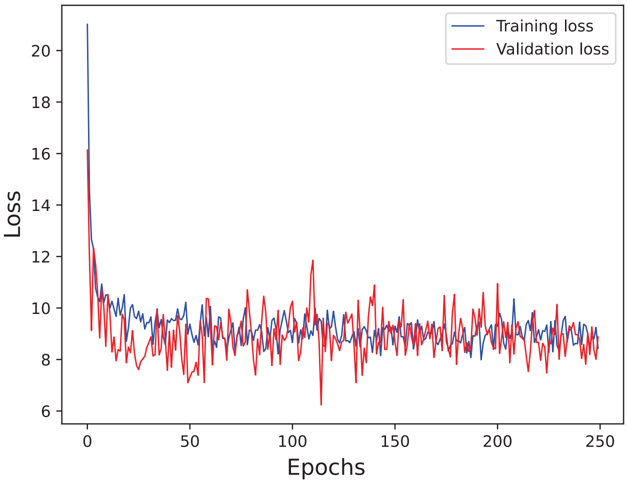

Fig. 10. Training and validation loss in one-shot settings using MAML

3) Fast adaptation: To analyze fast adaptation by the proposed meta learner, a test set was used that has never been seen during training the network. The optimized parameters during training meta-learner were taken as starting parameters for analyzing fast adaptation. Since,this is a new unseen task to adapt for the meta-learner, we only optimize the inner loop for adaptation to a new domain. We used ten inner steps for fast adaptation. Mean average error in predicting head poses in the unseen task was found out to be 7.33 in one-shot settings. Mean absolute error of 8.54, 8.64, and 4.83 was achieved for predicting yaw, pitch and roll, respectively. The optimized model generalizes well to an unseen set of tasks using only a few gradient updates.

Fig 11 shows the ground truth and predicted head pose angles in BIWI head pose dataset.

IV. Conclusions

This article proposed a method to use the meta-learning technique for the head pose estimation problem. The use of one-shot settings for training a meta learner encourages meta learning-based techniques for head pose estimation as they require less volume of labeled training data. Furthermore, implementation of representation learning before training a meta learner resulted in a more simpler network to predict head poses angles. The results show that the proposed method predicted correct Euler's angles with a mean average error (MAE_{avg}) of 7.33 in one-shot settings in the BIWI head pose dataset. The model successfully adapted to completely new, previously unseen test samples from the BIWI dataset and correctly predicted Euler's angles with only ten gradient descent updates. The results shows that the meta-learning based method can be used to effectively estimate head poses in few-shot settings.

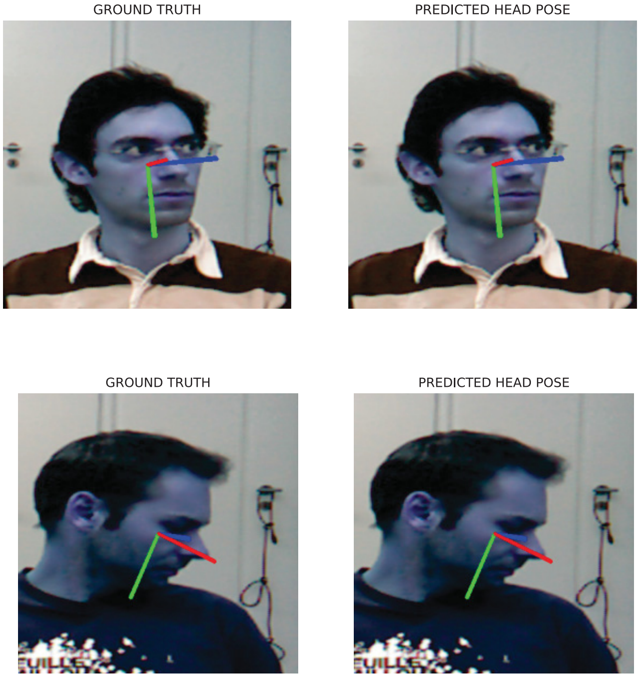

Fig. 11. Ground truth and predicted head pose angles. The red axis corresponds to the front of the face, while the green axis points down and the blue axis points to the side.

This work has room for lots of improvements. The more simplified methods such as Almost No Inner Loop (ANIL) [45], can be used instead of MAML to get improved performance in head pose estimation. Similar approach can be used to analyze meta-learning capabilities on other popular head pose datasets which can be structured in few-shot settings. Furthermore, the proposed method can find application in realtime head pose estimation in videos.

Meta-learning-based techniques has several limitations when we encounter diverse task distributions. In the real world, task distribution is often multi-modal, making it challenging for meta-learners to optimize. Furthermore, task families are required for meta-training using few-shot settings. Many head pose datasets doesn't have task families, making meta-learning based methods difficult to be used for head pose estimation.

References

[1] Luis Bergasa, José Buenaposada, Jesus Nuevo, Pedro Jimenez, and Luis Baumela. Analysing driver's attention level using computer vision. pages 1149 – 1154, 11 2008.

[2] Sileye O. Ba and Jean-Marc Odobez. Multiperson visual focus of attention from head pose and meeting contextual cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):101–116, 2011.

[3] David Gerónimo, Antonio M. López, Angel D. Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1239–1258, 2010.

[4] Rolf H. Baxter, Michael J. V. Leach, Sankha S. Mukherjee, and Neil M. Robertson. An adaptive motion model for person tracking with instantaneous head-pose features. *IEEE Signal Processing Letters*, 22(5):578–582, 2015.

[5] Dinh Tuan Tran and Joo-Ho Lee. A robust method for head orientation estimation using histogram of oriented gradients. In *Signal Processing, Image Processing and Pattern Recognition*, pages 391–400. Springer Berlin Heidelberg, 2011.

[6] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71, 06 2017.

[7] Jiao Bao and Mao Ye. Head pose estimation based on robust convolutional neural network. *Cybernetics and Information Technologies*, 16, 12 2016.

[8] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, jan 2019.

[9] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, 2019.

[10] Harry Frederick Harlow. The formation of learning sets. *Psychological review*, 56 1:51–65, 1949.

[11] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[12] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[13] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1126–1135. PMLR, 06–11 Aug 2017.

[14] Gabriele Fanelli, Thibaut Weise, Juergen Gall, and Luc Van Gool. Real time head pose estimation from consumer depth cameras. In *Pattern Recognition*, pages 101–110. Springer Berlin Heidelberg, 2011.

[15] J. Ng and Shaogang Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No.PR00378)*, pages 14–21, 1999.

[16] Jamie Sherrah, Shaogang Gong, and Eng-Jon Ong. Understanding pose discrimination in similarity space. In *Proceedings of the British Machine Vision Conference 1999, BMVC 1999, Nottingham, 13-16 September 1999*, pages 1–10, 1999.

[17] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[18] Patrick Burger, Martin Rothbucher, and Klaus Diepold. Self-initializing head pose estimation with a 2d monocular usb camera. 2014.

[19] R. Herpers, M. Michaelis, K.-H. Lichtenauer, and G. Sommer. Edge and keypoint detection in facial regions. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 212–217, 1996.

[20] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

[21] Qingshan Liu, Jing Yang, Jiankang Deng, and Kaihua Zhang. Robust facial landmark tracking via cascade regression. *Pattern Recogn.*, 66(C):53–62, jun 2017.

[22] Xin Jin and Xiaoyang Tan. Face alignment by robust discriminative hough voting. *Pattern Recogn.*, 60(C):318–333, dec 2016.

[23] Yongmin Li, Shaogang Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 300–305, 2000.

[24] S. Srinivasan and K.L. Boyer. Head pose estimation using view based eigenspaces. In *2002 International Conference on Pattern Recognition*, volume 4, pages 302–305 vol.4, 2002.

[25] Youding Zhu and K. Fujimura. Head pose estimation for driver monitoring. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 501–506, 2004.

[26] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2155–215509, 2018.

[27] Ben Benfold and Ian Reid. Colour invariant head pose classification in low resolution video. In *Proceedings of the 19th British Machine Vision Conference*, September 2008.

[28] Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *CVPR 2011*, pages 617–624, 2011.

[29] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, Oswald Lanz, and Nicu Sebe. A multi-task learning framework for head pose estimation under target motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1070–1083, 2016.

[30] Margarita Osadchy, Yann Le Cun, and Matthew L. Miller. *Synergistic Face Detection and Pose Estimation with Energy-Based Models*, pages 196–206. Springer Berlin Heidelberg, 2006.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Byungtae Ahn, Jaesik Park, and In So Kweon. Real-time head orientation from a monocular camera using deep neural network. In *Computer Vision – ACCV 2014*, pages 82–96, Cham, 2015. Springer International Publishing.

[33] Sankha S. Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11):2094–2107, 2015.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[35] Marco Venturelli, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Deep head pose estimation from depth data for in-car automotive applications, 2017.

[36] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71, 06 2017.

[37] Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations*, 2019.

[38] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9367–9376, 2019.

[39] Tadas Baltrusaitis. *Automatic facial expression analysis*. PhD thesis, 04 2014.

[40] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.

[41] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[42] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[43] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[44] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020.

[45] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *ArXiv*, abs/1909.09157, 2020.