International Journal of Embedded and Real-Time Communication Systems
Volume 14 • Issue 1

One, Five, and Ten-Shot-Based Meta-Learning for Computationally Efficient Head Pose Estimation

Manoj Joshi, Institute of Engineering, Nepal*
https://orcid.org/0000-0002-6832-0874
Dibakar Raj Pant, Institute of Engineering, Nepal
Jukka Heikkonen, University of Turku, Finland
Rajeev Kanth, Savonia University of Applied Sciences, Finland
https://orcid.org/0000-0003-1109-1211

ABSTRACT

Many real-world applications rely on head pose estimation. The performance of head pose estimation has significantly improved with techniques like convolutional neural networks (CNN). However, CNN requires a large amount of data for training. This article presents a new framework for head pose estimation using computationally efficient first-order model-agnostic meta-learning (FO-MAML)-based method and compares the performance with existing MAML-based approaches. Experiments using one-shot, five-shot, and ten-shot settings are done using MAML and FO-MAML. A mean average error (MAE_{avg}) of 7.72, 6.30, and 5.32 has been achieved in predicting head pose using MAML for one-, five-, and ten-shot settings, respectively. Similarly, MAE_{avg} of 8.33, 6.84, and 6.23 has been achieved in predicting head pose using FO-MAML for one-, five-, and ten-shot settings, respectively. The computational complexity of an outer-loop update in MAML is found to be $O(n^2)$ whereas for FO-MAML it is O(n).

KEYWORDS

Convolutional Neural Networks, Deep Learning, Face Detection, Few-Shot Learning, Head-Pose Estimation, Meta-Learning, Neural Networks, Optimization

INTRODUCTION

In the last few years, significant advancement has been seen in the area of computer vision, robotics, and human-machine interaction. With increasing areas of applications in gaze estimation, self-driving cars, and impaired assistance, a reliable head pose estimation framework has been important. Prior

DOI: 10.4018/IJERTCS.316877
*Corresponding Author

research has been done on head pose estimation for understanding how human attention works (Bergasa et al., 2008). It also fits in applications such as analyzing human behavior and social interactions (Ba & Odobez, 2011). Head pose estimation becomes crucial in driver assistance systems to slow down the vehicle when pedestrians are not aware of the presence of the vehicle (Geronimo, López, Sappa & Graf, 2010). Because of this significance, head pose estimation has been thoroughly investigated and explored in various fields.

Head-pose estimation plays a prominent role in use cases such as anomaly detection, surveillance, human-computer interface (HCI), and understanding behavioral dynamics in the crowd (Baxter, Leach, Mukherjee & Robertson, 2015). The extreme facial orientations, varying illumination and resolution, makeup, and presence of hairs in the human face make it challenging to predict head pose. Traditional methods gained some success in head pose estimation using image processing techniques. Histogram of Oriented Gradients (HOG) methods successfully predicted head poses in images and videos (Tran & Lee, 2011). The traditional methods for head pose estimation were founded on discriminative/landmark-based or parameterized appearance-based models. The traditional approaches worked well in estimating head pose but were not flexible and robust to extreme variation in the head pose.

The development of convolutional neural networks (CNN) became a popular choice for estimating head poses (Patacchiola & Cangelosi, 2017) because of their high efficiency. The efficiency of CNNs is reliant on the amount of well-annotated data samples. The more annotated data we have, the more efficiently CNN will perform. But capturing a large and well-annotated dataset is difficult in most cases. Convolutional neural networks while using a large volume of data, are good at predicting head poses, although they lack generalization. A good head poses estimator should be data efficient and have similar efficiency as that of the CNNs. It should also adapt to unseen faces and perform much better as more and more evidence of head pose features becomes available.

In recent years, few-shot learning techniques have been more popular when less data is available. Meta-learning-based techniques gained popularity in the past few years, as they can be applied in few-shot settings and adapt well to unseen data (Sun, Liu, Chua & Schiele, 2018). These techniques can use the knowledge gained from previous experiences and use it to boost their future performance. Meta-learners can learn a novel task from a limited training dataset, and use it to generalize to unseen tasks that the model encounters in the future. This learning method is called learning-to-learn. The use of meta-learning can benefit us with better data and computational efficiency.

This article extends the work (Joshi, Pant, Karn, Heikkonen & Kanth, 2022). The article revises the existing MAML based approach and then proposes a novel approach of using computationally efficient first-order model-agnostic meta-learning (FO-MAML). The novel approach performed well in head-pose estimation and is computationally more efficient. One, five, and ten-shot based experiments have been performed in BIWI head pose dataset using MAML and FO-MAML and comparison has been made in terms of accuracy and time complexity of both approaches.

BACKGROUND

Conventional techniques for head pose estimation used appearance templates. Ng and Gong (1999) extended support vector machines (SVM) and modeled the appearance of human faces in 2D. Their work gave excellent results in multi-view face detection and head pose estimation. Sherrah et al., (1999) applied Gabor filters to improve the pose difference at each pose angle. They applied principal component analysis (PCA) to get the identity-invariant properties of human faces. The identity-invariant features were used to compute the head poses. The success of frontal face detection using supervised learning algorithms gave researchers a new direction to find methods for estimating head poses. Researchers extended single-face detector methods to multiple-face detector methods. Hence, detector array-based methods have been developed (Osuna, Freund, & Girosit, 1997). In these techniques, a detector is trained with multiple images rather than comparing facial images to some

predefined templates. Mostly SVMs were implemented in detector array-based methods and were successful in predicting head poses.

Geometric models were later developed that used facial key points for computing head pose. Using facial key points, we don't need to have previous knowledge of the user's facial appearance or orientation (Jin & Tan, 2016). Such a method allows us to detect facial features based on anatomical landmarks of the face. However, it requires accurate edge and line detection of human facial key points. Sun et al., (2013) suggested an approach using convolutional neural networks (CNN) for predicting the positions of facial key points. The CNN locates anatomical facial landmarks with high accuracy from the entire face. It also estimates all the key points simultaneously. Liu et al., (2016) did similar work on sequential images. These methods gave an excellent result in computing head poses, although it is exceptionally tedious to detect facial landmarks in human faces. Kashevnik et al., (2021) worked on detecting head pose angles in the context of driver monitoring application using facial landmarks and showed their method have superior performance than the state-of-the-art when pose varies extremely. The authors thoroughly experimented and showed their method could detect head pose angles when the rotation of the head is up to 70º. Srinivasan and Boyer (2002) studied view-based eigenspaces to estimate head poses. They claimed that eigenspaces of facial images were more efficient for accurate facial recognition, which can improve head pose estimation. Their method was more powerful and computationally less expensive than the existing methods. Zhu and Fujimura (2004) proposed a method based on PCA and 3D motion estimation. They further used the depth information of facial images to segment human faces even in a cluttered environment.

Few classification-based methods were developed using discretized head poses. Benfold and Reid (2008) used an ensemble of the tree-based algorithm to detect color-invariant head pose. Fanelli et al., (2011) used depth data for estimating head pose using random forest regression. Yan et al., (2016) have proposed a multi-task learning approach to calculate the head poses of mobile persons in an environment observed by multiple surveillance cameras. Enhancements in deep learning led to the development of non-linear regression-based head pose estimation methods. Ruiz et al., (2018) proposed a method that did not need to use facial key points for estimating head pose. They trained a multi-loss convolutional neural network (CNN) for predicting head pose angles. A real-time CNN-based approach that is similar to LeNet-5 (Lecun et al., 1998) has been suggested to predict head pose (Osadchy, Miller & Lecun, 2004). Their CNN-based architecture has more feature maps than LeNet-5. Other CNN-based methods used RGB images along with depth images. Ahn et al., (2015) used GoogleLeNet (Szegedy et al., 2014) to estimate head poses using RGB and depth information. Venturelli et al., (2017) proposed a lightweight model having five convolution layers and three dense layers and achieved better performance. Recently, the work of Patacchiola and Cangelosi (2017) has shown the better performance of CNN in estimating head pose using multiple datasets.

CNN-based methods give excellent results in estimating head pose. The demerit of using CNN is that CNN requires huge training data to find the best estimator and lacks generalization ability on new data. CNN is not good enough to adapt to a novel set of tasks. A model-independent meta-training approach was introduced that can make use of gradient descent for training (Finn, Abbeel, & Levine, 2017). Model-agnostic meta-learning (MAML) can be used in any model that uses gradient descent for training. Sun et al., (2019) suggested an approach to use a meta-learning technique in few-shot settings. Their approach showed that meta-learning performs admirably in few-shot learning. The fundamental concept of their work was to train a base learner with a small number of examples randomly selected from a set of tasks from a distribution. Then, the base learner is adapted to learn new tasks using identical examples. Antoniou et al., (2019) have shown that model-agnostic meta-learning (MAML) can be used to solve regression and classification problems. Park et al., (2019) have worked on meta-learning using (MAML) and suggested an architecture to estimate gaze in few-shot settings. For training a person-specific gaze estimation model using meta-learning, they utilized a small number of calibration samples. Their approach also allows for a far more accurate estimate of

gaze direction for any new person. Few-shot learning for head pose estimation has been studied by Joshi et al., (2021). They used ResNet (He et al., 2016) as a feature extractor and trained a few-shot learner to estimate head poses. Joshi et al., (2022) suggested a model-agnostic meta-learning (MAML) based framework for head pose estimation. Their method gave better results in estimating head pose using MAML in few-shot settings.

METHODOLOGY

In this article, the authors extend the work of Joshi et al., (2022) by omitting the second-order gradients during model-agnostic meta-learning training. A three-stage architecture has been introduced consisting of Face Detection Framework, Representation Learning Framework, and First-Order Meta-Learning Framework. The architecture for head poses estimation using meta-learning is shown in Figure 1. The next sections explain each stage of head pose detection.

Dataset

For this work, the popular BIWI head pose dataset has been used to train the proposed meta-learner network (Fanelli, Weise, Gall, & Van Gool, 2011). This head poses dataset consists of 15,678 images of 20 different subjects in varying environmental conditions. Each image is of size 640 x 480 pixels. Along with RGB images of the subjects, a depth image corresponding to it having a size of 640 × 480 pixels is given. The annotated head pose angles are also provided with the dataset. The head poses of the subjects in the dataset vary approximately ±75∘ in yaw, ±60∘ in pitch, and ±50∘ in the roll. The ground truth for the 3D position and rotation of the heads has also been provided with the images of the subjects.

Experimental Testbench

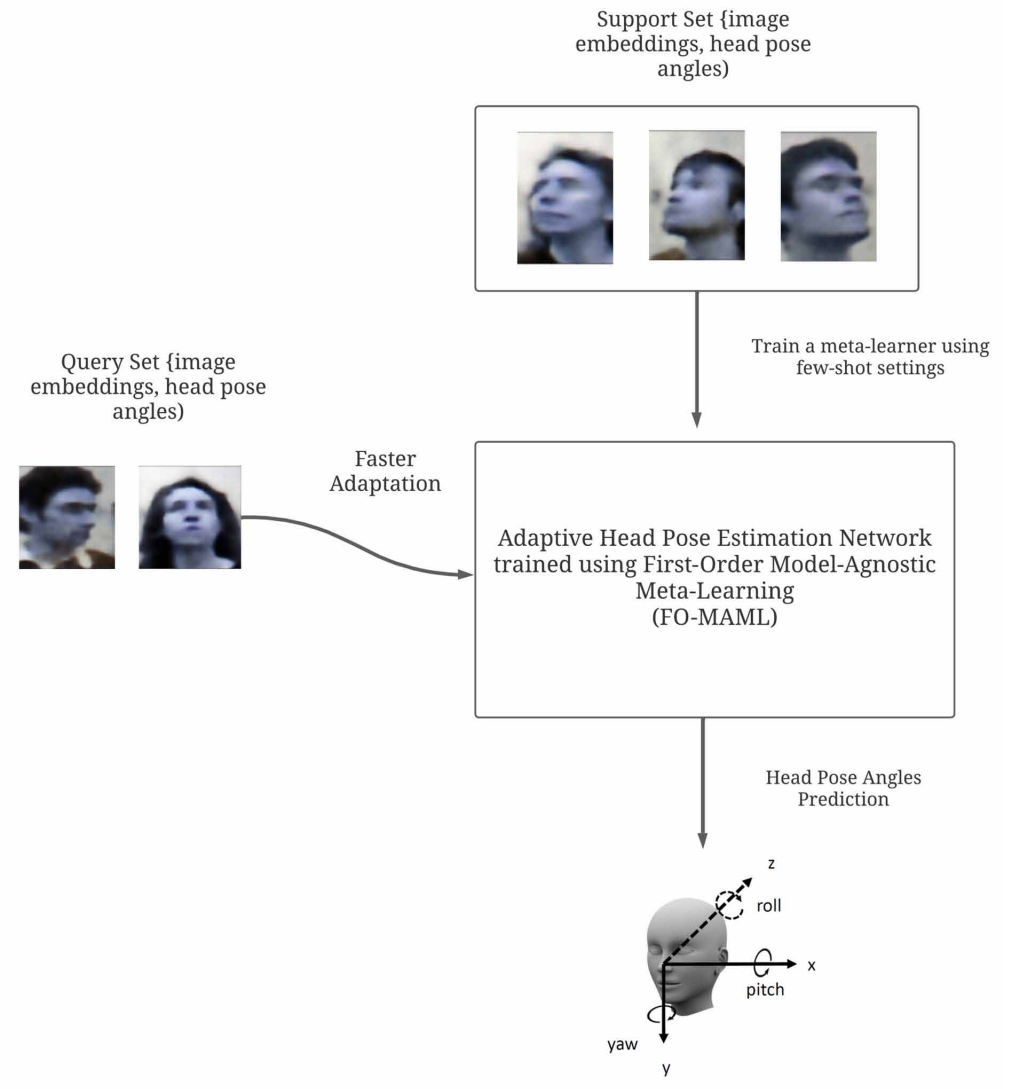
PyTorch and PyTorch Lightening libraries have been used to create variational autoencoder and meta-learner models. For hyperparameter tuning and neural architectural search, Ray Tune library has been used (Liaw et al., 2018). The meta-training has been done on the free Google Colab platform. The training utilizes Google Colab's free NVIDIA Tesla K80 GPU with 12GB memory. For computing the second-order gradient a library called Higher has been used with PyTorch Lightening (Grefenstette et al., 2019). For image processing, OpenCV library has been used for image processing where needed.

Data Preprocessing

Some images in the BIWI dataset consist of multiple background objects. Hence, we have to identify and crop the human faces from the images. Face detection requires proper face localization and bounding box coordinates prediction. For this work, we use a multi-task cascaded convolutional neural network (MTCNN) for detecting faces and predicting bounding box coordinates (Zhang et al., 2016). After applying MTCNN, we crop the identified faces from BIWI images to a size of 128 × 128. The dataset created from cropped images is then split into two: the first fifteen subjects form a distribution from which few-shot tasks can be randomly sampled for training the meta-learner, whereas the remaining five subjects were used during testing the meta-learner. A total of 10581 cropped faces have been used as a distribution to sample training tasks, and the remaining 2638 faces were used for testing. When detecting the faces, images with heavily veiled backgrounds, numerous objects, and extreme stances were eliminated.

The faces obtained from MTCNN were normalized before passing into the representation learning framework. After normalizing with min-max normalization, we get pixels in between 0 and 1. Min-max normalization bound the pixel values into a smaller standard deviation which works well in presence of outliers. Min-max normalization is computed as,

Figure 1. Illustration of head pose estimation using First-Order Model-Agnostic Meta-Learning architecture



$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x is the value of a pixel, x_{min} and x_{max} is the minimum and maximum value of the pixels.

Representation Learning Framework

This article presents a representation learning framework designed to learn important features from human faces in latent space. There can be many methods for representation learning, but variational autoencoders (VAE) seem excellent for these sorts of tasks. Furthermore, VAE networks can link representation learning with generative modeling making it possible to generate usable data from scratch. Based on this, a variational autoencoder network has been implemented for this task. The

network learns the relevant characteristics of the human faces, thus producing valuable latent embeddings. The architecture of the variational autoencoder (VAE) is shown in Figure 2. An encoder and decoder are defined as $E : x \rightarrow z$ and $D : z \rightarrow \hat{x}$ respectively, such that $D\left(E\left(x\right)\right) = \hat{x}$. If x is an input to the network, an encoder encodes it into latent variable $\hat{x}$. In this article, a VAE is implemented to learn the characteristic features from training images and produce a latent space of 200 features. The size of the latent space has been decided by computing the reconstruction loss repeatedly by varying latent dimensions z. The number of latent dimensions that give the least combined error (KL-Divergence and Mean-Squared Error) while decoding is selected as the size of the latent embedding vector.

The encoder network contains five convolution layers. Each convolution layer uses LeakyReLu activation function. A stride of size two and a dropout rate of 0.25 has been taken into account. The first four convolutional layers have been implemented using the kernel size of three. The last convolutional layer uses a kernel of size one. The outputs of the convolutional layers are flattened to obtain 200 latent embedding vectors. The latent embedding vectors are then sent to a liner layer which performs a linear transformation on it to generate 4096 latent features.

The decoder comprises four transposed convolution layers with the LeakyReLu activation function. The stride of two and kernel size of three have been considered for transposed convolution layer. A dropout rate of 0.25 has been taken. The reconstructed images of size 128 x 128 are produced from the last transposed convolution layer. The size of the kernel in the encoder and decoder, dropout rate, number of filters, and number of convolutional layers has been searched as a part of hyperparameter tuning.

Meta-Learning Framework

In the meta-learning framework, we train an adaptive head pose estimator that finds the optimal parameters for a given training dataset and generalizes well to unseen data. For achieving this, the authors have implemented optimization-based meta-learning approaches such as MAML and FO-MAML and compared their performance in estimating head poses. Figure 3 shows the differences between MAML and FO-MAML-based architecture.

Figure 3a shows a MAML-based approach for a single task. We can see that computing meta-gradient in MAML requires computation all the way up the computation graph. MAML requires

Figure 2. The architecture of a variational autoencoder

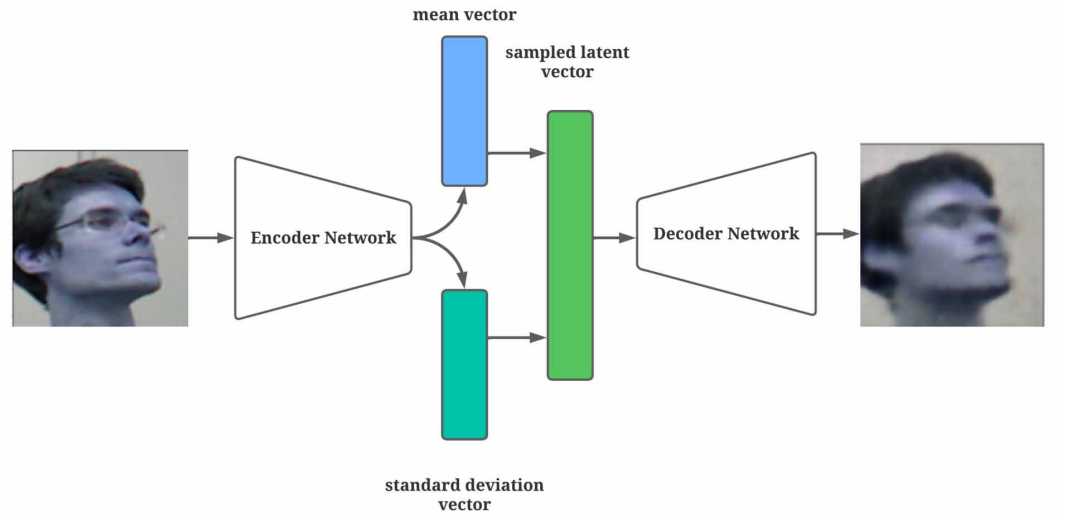
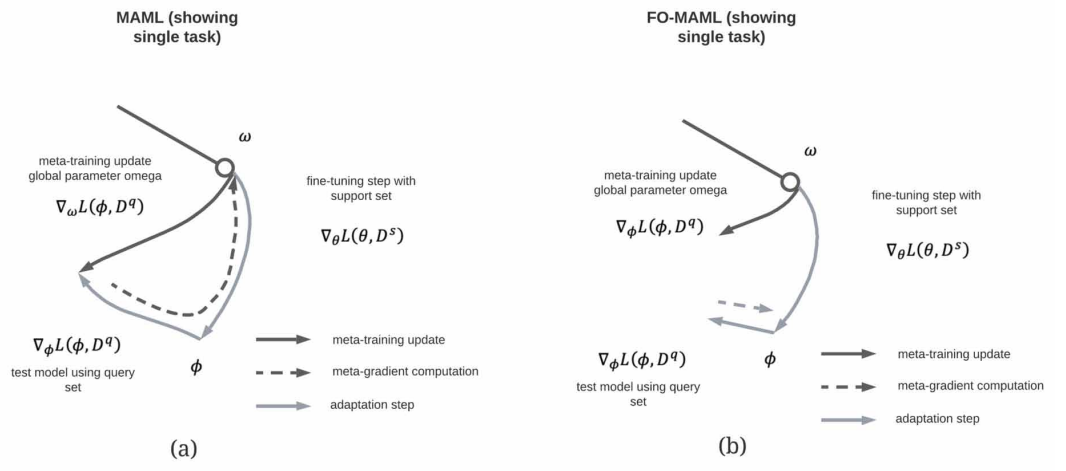


Figure 3. The architecture of a MAML and FO-MAML



calculating the derivative of the loss function $L\left(\phi, D^q\right)$ while updating the parameter ω. It means we need to compute the Hessian matrix given by $\nabla_\omega L\left(\phi, D^q\right)$ during the meta-learning step. A computationally more efficient approach compared to MAML is FO-MAML as shown in Figure 3b. In FO-MAML, we are using the first-order approximation $\nabla_\phi L\left(\phi, D^q\right)$ for updating global parameter ω. This approach removes the need for computing Hessian. This allows us to omit the second-order gradients during learning making the optimization of the meta-learner faster.

Model-Agnostic Meta-Learning (MAML)

MAML is a model-agnostic and task-agnostic approach that can swiftly train model parameters for quick adaption to new tasks. The benefit of MAML based approach is in learning new tasks rapidly with only a few gradient modifications in the model.

Algorithm 1: Generic Model-Agnostic Meta-Learning Algorithm
Require: $D\left(\tau\right)$: distribution over tasks (human faces)
Require: α, β: learning rates
Step 1. Randomly initialize parameter ω
Step 2. while not done do
Step 3. Sample batch of task $T_i \sim D\left(\tau\right)$
Step 4. For all T_i do
Step 5. Evaluate $\nabla_\omega L\left(\omega\right)$ with respect to K samples
Step 6. Calculate adapted parameters using gradient descent:

$$\omega_i = \omega - \alpha \nabla_\omega L\left(\omega\right)$$

Step 7. end for
Step 8. Update $\omega \leftarrow \omega - \beta \nabla_\omega L\left(\omega - \alpha \nabla_\omega L\left(\omega\right)\right)$
Step 9. end while

Let us consider a model f_{ω} having parameters ω with τ tasks, as shown in Algorithm 1. With just K samples taken at a time, the model $f_{\dot{E}}$ is trained with tasks T_i selected from the distribution $D\left(\tau\right)$. This produced a reliable few-shot learner, that can generalize well to novel examples taken from the new task T_i which is again selected from the distribution $D\left(\tau\right)$. Our objective is to minimize loss L_{τ_i} for task T_i, hence the model f_{ω} need to be updated regularly.

The proposed head pose estimator uses MAML and FO-MAML based model. Let M be the head pose estimator model that learns optimal parameters ω^* after training with meta-learning. The model is said to be optimized once it learns the optimal parameters ω^*. This would result in a fine-tuned model M_{ω}^* that can generalize pretty well to unseen sets of tasks sampled from the distribution $D\left(\tau\right)$. We create a sample set consisting of support set and query set from training and testing. Let us denote the support set and the query set for training as $\left\{S_s^{train}, S_q^{train}\right\}$ selected from the distribution $D\left(\tau\right)$. Similarly, the support set and the query set for testing is put together as $\left\{S_s^{test}, S_q^{test}\right\}$. The random sampling strategy has been used to sample the support and the query set from the distribution $D\left(\tau\right)$.

The support and the query sets consist of $\left\{\left(z, gt\right)\right\}$, where z is the latent representation of the input training sample and gt is the ground truth of the head pose angles. We generally select small number of examples $\left(\leq 20\right)$ in the support and the query set for training in the few-shot scenario. A cost function for the gradient update is the mean absolute error (MAE) in estimating yaw, pitch, and roll. Meta-learning computes the loss for support set S_s^{train} and updates the weights ω_t at step t using a few gradient steps. An inner-learning rate α is used for computing the new gradients. A person p^{train} is selected from S_s^{train} and parameter ω is updated as shown below.

$$\omega_t^{'} = f\left(\omega_t\right) = \omega_t - \alpha \nabla L_{p^{train}}^{s}\left(\omega_t\right) \tag{2}$$

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

$$L = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|gt_i - \widehat{y_i}\right| \tag{3}$$

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 $\widehat{y_i}$ are the predicted head poses angles.

Using these updated weights $\omega_t^{'}$, we now compute loss for validation set S_q^{train} at step $t+1$. 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.

$$\omega_{t+1} = \omega_t - \beta \nabla L_{p^{train}}^{q}\left(f\left(\omega_t\right)\right) \tag{4}$$

The meta-gradient updates continue until the model finds optimal weights ω^*. After we achieve optimal parameters ω^*, we can use the model M_{ω}^* to generalize to unseen examples from sets

$\left\{S_s^{test}, S_q^{test}\right\}$. Sample a person p^{test} from the test set S_s^{test} to fine-tune and adapt our model M_ω^* as shown below.

$$\omega_p^{test} = \omega^* - \alpha \nabla L_{p^{test}}^s \left(\omega^*\right) \tag{5}$$

Finally, we verify fast adaptation using sample images sampled from the validation set S_q^{test}.

Joshi et al., (2022) suggested an approach to estimate accurate head poses using MAML, however, a significant computational expense comes with MAML since it computes second-order derivatives during the backpropagation of meta-gradients. Hence, we implement a first-order approximation of the generic MAML algorithm that eliminates the second derivatives during backpropagating the meta-gradients through inner loop updates. Despite the absence of second-derivative terms, the resulting technique is able to compute the meta-gradients. Interestingly, this method performs identically to that obtained by considering second-order derivatives. This implies that the gradients of the model at the post-update parameter values are responsible for the improvement in MAML and the second-order updates are less significant.

First-Order Model-Agnostic Meta Learning (FOMAML)

Algorithm 2 shows the first-order model-agnostic meta-learning algorithm step by step. All the training process remains similar to MAML except for updating the parameters at time $t+1$. After we get the updated parameter $\omega_t^{'}$ at time t, the new gradients at time $t+1$ are then computed with respect to the newly updated parameter $\omega_t^{'}$ using β as the learning rate. Finally, the parameters ω_t are updated with the objective of minimizing the validation loss, as shown below.

$$\omega_{t+1} = \omega_t^{'} - \beta \nabla L_{p^{train}}^q \left(\omega_t^{'}\right) \tag{6}$$

Algorithm 2 show the first-order model-agnostic meta-learning (FO-MAML) algorithm in detail.

Algorithm 2: Generic First-Order Model-Agnostic Meta-Learning Algorithm

Require: $D\left(\tau\right)$: distribution over tasks (human faces)

Require: α, β: learning rates

Step 1. Randomly initialize parameter E

Step 2. while not done do

Step 3. Sample batch of task $T_i \sim D\left(\tau\right)$

Step 4. For all T_i do

Step 5. Evaluate $\nabla_\omega L\left(\omega\right)$ with respect to K samples

Step 6. Calculate adapted parameters using gradient descent:

$$\omega_i = \omega - \alpha \nabla_\omega L\left(\omega\right)$$

Step 7. end for

Step 8. Update $\omega \leftarrow \omega - \beta \nabla_{\omega_i} L\left(\omega_i\right)$

Step 9. end while

Mathematical Formulation for MAML and its First-Order Approximation (FOMAML)

We are particularly interested in comparing the meta-objectives between FO-MAML and MAML. For this let us look at the MAML optimization problem: find some initial parameters, ω, such that the learner will have minimum loss L_τ after k SGD updates for a randomly selected task τ. Let us define few notations for this formulation:

$$U_\tau^k \rightarrow k \; SGD \; updates \; on \; examples \; sampled \; from \; task$$

$$L_\tau \rightarrow Loss \; function \; \left(Mean \; Absolute \; Error\right) \; on \; selected \; task$$

Then we can define MAML update rule as follows for $k number of gradient steps$:

$$\min imize_{\omega E_\tau} L_\tau \left\{U_\tau^k\left(\omega\right)\right\} \tag{7}$$

For computing this update, we actually compute the gradient with respect to our parameters ω. For simplicity, let us omit k, because this is constant for our computation.

$$g_{MAML} = \frac{\partial}{\partial\omega}\left[L_\tau\left(U_\tau\left(\omega\right)\right)\right] = U_\tau^{'}\left(\omega\right)L_\tau^{'}\left(\omega_{new}\right) \tag{8}$$

where $\omega_{new} = U_\tau\left(\omega\right)$ and $U_\tau^{'}$ is the Jacobian matrix of the MAML update operation with respect to initial parameters ω. Adding the sequence of the gradient vectors in each gradient steps to the initial vector result in $U_\tau\left(\omega\right)$. First-Order MAML (FO-MAML) assumes these gradients to be constant, thus, replacing the Jacobian $U_\tau^{k'}$ by the identity operation. Thus, the gradients used by FO-MAML for outer-loop optimization is given by:

$$g_{FO-MAML} = L_\tau^{'}\left(\omega_{new}\right) \tag{9}$$

These gradients can directly be used in outer-loop optimization in FO-MAML given by:

$$\omega \leftarrow \omega - \beta g_{FO-MAML} \tag{10}$$

From these mathematical formulations, the FO-MAML can be efficiently implemented as shown in Algorithm 3.

`Algorithm 3: Gradient update in First-Order Model-Agnostic Meta-Learning`

`Step 1. Sample task` τ `from distribution` D_τ

`Step 2. Apply the update operator to yield` $\omega_{new} = U_\tau\left(\omega\right)$

`Step 3. Compute the gradients at` ω_{new}`:`

$$g_{FO-MAML} = L_\tau^{'}\left(\omega_{new}\right)$$

Step 4. Plug computed gradient to update outer loop:

$$\omega \leftarrow \omega - \beta g_{FO-MAML}$$

Meta-Learner Network

A multi-layer fully connected neural network is implemented as a meta-learner. 200 latent embeddings are taken as input to the linear layer and produce 1000 features after linear transformation. The features are then passed through three dense layers each of which gives 1000 features. The final layer of the network comprises of fully connected linear layer and regresses three vectors as head poses angles. LeakyReLu has been taken as an activation function with a negative slope of 0.1. Similarly, a dropout rate of 0.25 is considered. This regression model forecasts the yaw, pitch, and roll angles. Figure 4 depicts the proposed architecture for meta-training. The number of output features, number of dense layers, dropout rate, and activation function have been chosen as a part of hyperparameter tuning.

Hyperparameter Tuning

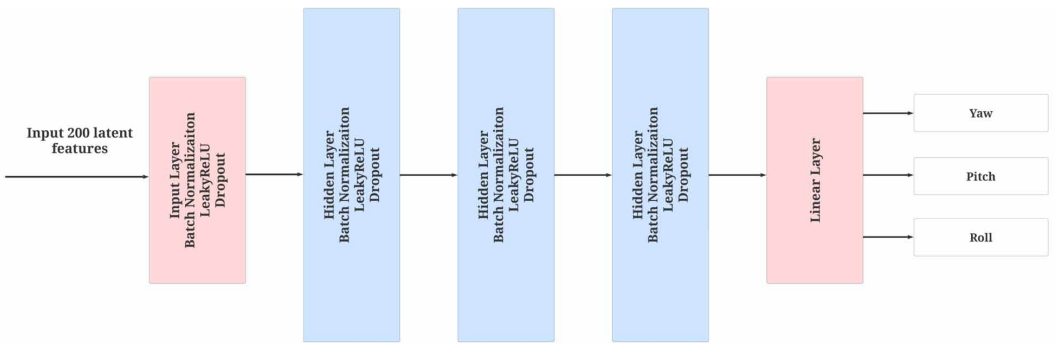
A grid search-based approach has been used to find optimal hyperparameters of both a meta-learner and a variational autoencoder as a part of hyperparameter optimization. For this, Ray Tune library has been taken into consideration. Using grid search, the meta-learning hyperparameters and were chosen 0.01 respectively. Similarly, a learning rate of 0.001 has been taken for training a variational autoencoder. Ten inner gradient steps have been considered to train MAML and FO-MAML. The dropout rates of 0.25 have been taken for both meta-learner and VAE. For training the meta-learner, stochastic gradient descent (SGD) has been used whereas Adam is used as an optimizer for VAE. Other hyperparameters such as the number of dense layers, convolutional layers, and the number of features in between dense layers have been searched using Ray Tune.

Model Evaluation

The pixel-wise mean squared error (MSE) is computed during representation learning. The squared error between actual and regenerated faces has been used to evaluate the performance of VAE. For computing the similarity between actual and predicted head poses, mean absolute error (MAE) has been used. The mean absolute error (MAE) between actual and predicted head poses is given in equation 11.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y_i}\right| \tag{11}$$

Figure 4. Meta-learner architecture



where y_i are the ground truth head pose angles and $\widehat{y_i}$ is the predicted head pose angles.

The average of three mean absolute errors in predicting Euler's angles is taken as the finalscore 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:

$$MAE_{avg} = \frac{yaw_{mae} + pitch_{mae} + roll_{mae}}{3} \tag{12}$$

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

RESULT AND ANALYSIS

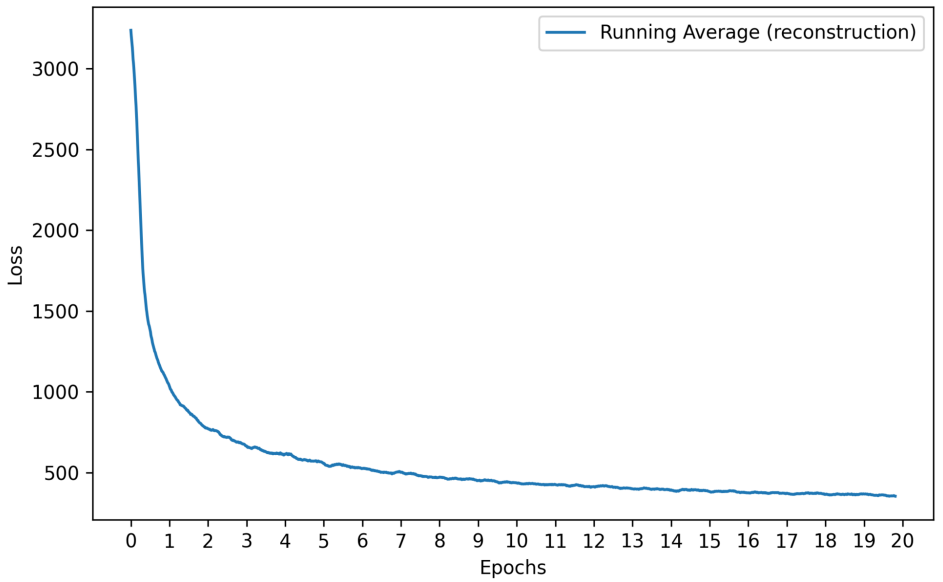
In this section, we analyze the results of the proposed representation learning and meta-learning framework. We present the results from one, five, and ten-shot experiments using MAML and FO-MAML-based approaches for comparing the performance of both methods.

Representation Learning

A variational autoencoder (VAE) is trained to generate latent embedding that preserves the head pose. For this, facial images of size 128×128 are used for training a VAE. In this article, a VAE is trained for 20 epochs to generate 200 latent embeddings. The pixel-wise mean squared loss is computed for the performance evaluation of VAE. Figure 5 shows the pixel-wise reconstruction loss of a VAE while regenerating original faces using latent embeddings. The mean squared error loss during reconstruction is decreasing with further training.

The Kullback–Leibler divergence (KL-Divergence) loss during reconstruction is shown in Figure 6. During training, reconstruction loss is given a higher privilege than KL-Divergence loss. Hence,

Figure 5. Mean squared error loss during reconstruction using a VAE



MSE loss is large at the start and decreases gradually with training whereas KL-Divergence loss is lower initially but increases with further training (Asperti & Trentin, 2020). The total loss of a VAE is balanced after lowering the MSE loss and increasing the KL-Divergence loss with a few epochs of training.

Reconstruction loss and KL-Divergence loss combined together to form the total loss of a VAE which is shown in Figure 7. The embedding produced from a VAE is used to train a meta-learner. Figure 8 shows the original and reconstructed facial images using a VAE.

Figure 6. KL-Divergence Loss during reconstruction using a VAE

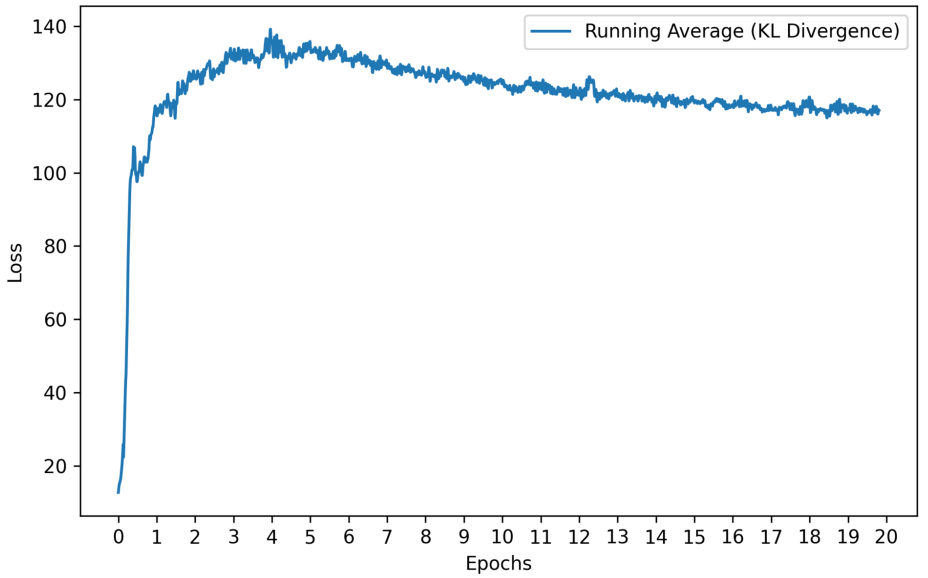


Figure 7. The total combined loss of a VAE

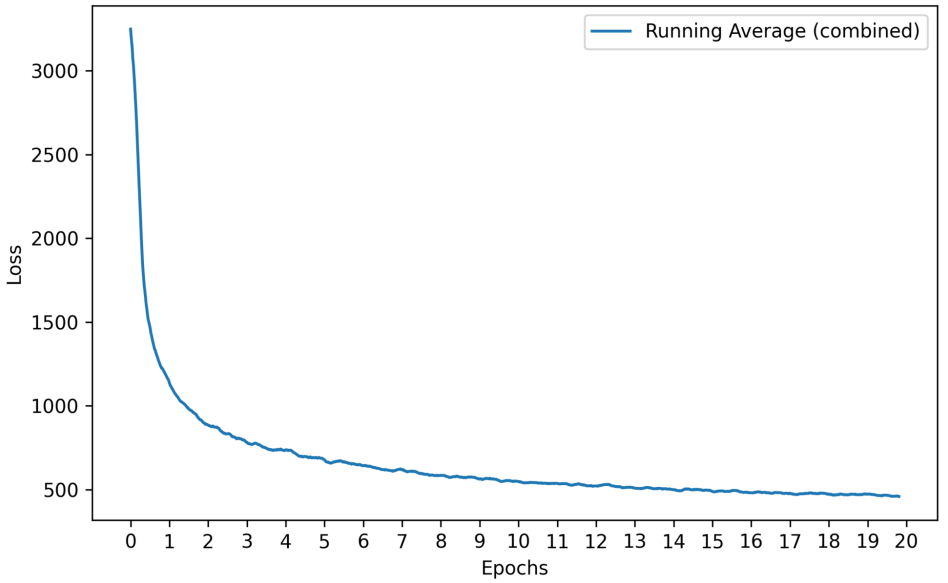
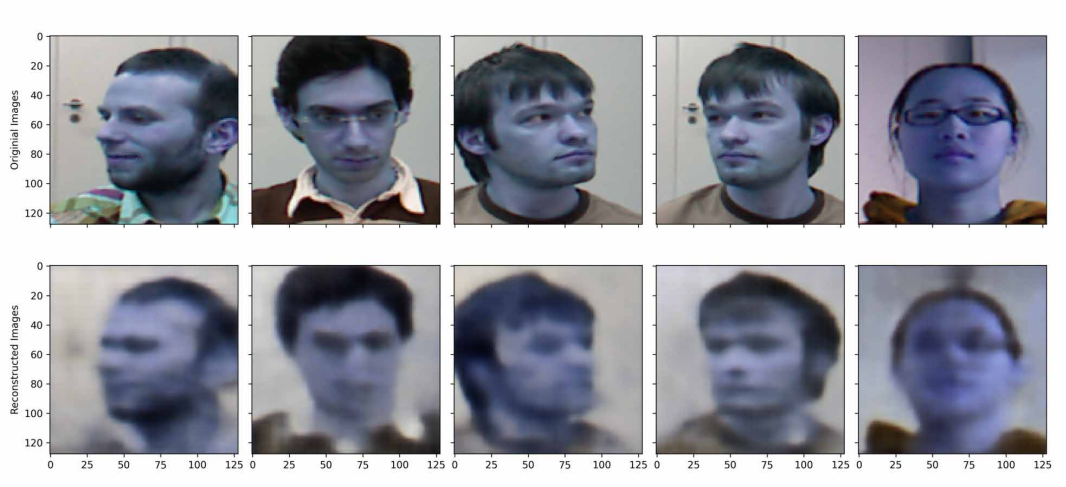


Figure 8. Original and reconstructed faces using a VAE



One-Shot Settings Using MAML and FO-MAML

To construct a support and query set for MAML and FO-MAML training in one-shot scenarios, one sample from each of five different individuals is randomly chosen. The query set is used to fine-tune the network parameters while the support set is utilized for training. Mean absolute error (MAE) has been used to evaluate the performance of the model in predicting head pose angles. We successfully predicted the head pose angles with a mean average error $\left(\mathrm{MA}E_{avg}\right)$ of 7.72 using MAML. For the experiment in one-shot settings, we trained the meta-learner for 250 episodes with a meta batch size of 64. The training and validation loss of meta-learner using MAML in the one-shot setting is shown in Figure 9.

Similarly, we successfully predicted the head pose angles with a MAE_{avg} of 8.33 using FO-MAML. For experimenting with using FO-MAML in one-shot settings, we trained the meta-learner for 250 episodes with the meta batch size of 64. Stochastic Gradient Descent (SGD) is used as an optimizer during meta-learning.

The training and validation loss of meta-learner using FO-MAML in the one-shot settings is shown in Figure 10. Smoother results have been obtained in the case of MAML. This is because we are adding gradient vectors in each update step making use of second-order gradients during backpropagation. The results also show that there is almost similar performance when we ignore the second-order gradients during the meta-learning training. For comparing the time taken to update the meta-gradient in both MAML and FO-MAML-based methods, we computed the time taken for a single outer-loop update. The MAML approach took about $7.15\mu\ sec$ for a single outer-loop update whereas it only took $5.41\mu sec$ for FO-MAML.

Five-Shot Settings Using MAML and FO-MAML

Five samples from each of five different individuals are randomly chosen for creating support and query sets for training MAML and FO-MAML in five-shot scenarios. Mean absolute error (MAE) has been used to evaluate the performance of the model in predicting head pose angles. In five-shot settings, the head pose angles have been predicted with a MAE_{avg} of 6.30 using MAML. For the experiment in five-shot settings, a similar setup is used as that of one-shot settings i.e., episode =

Figure 9. Training and validation loss in one-shot settings using MAML

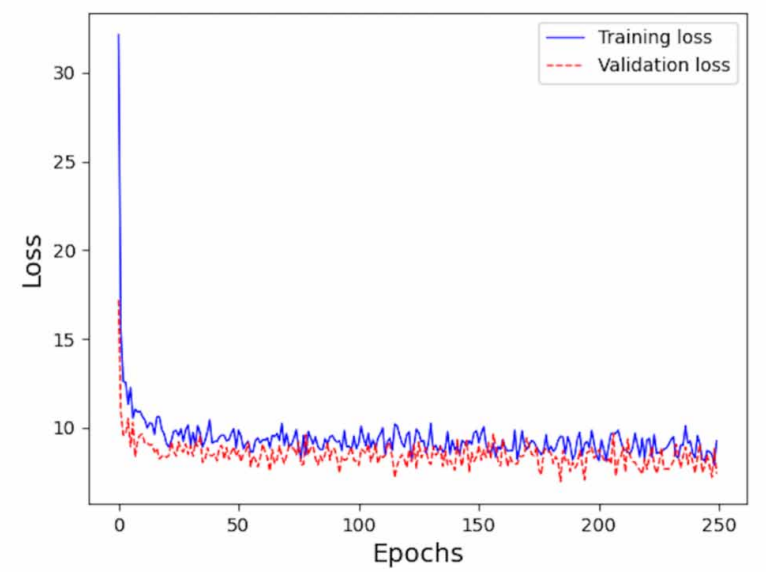
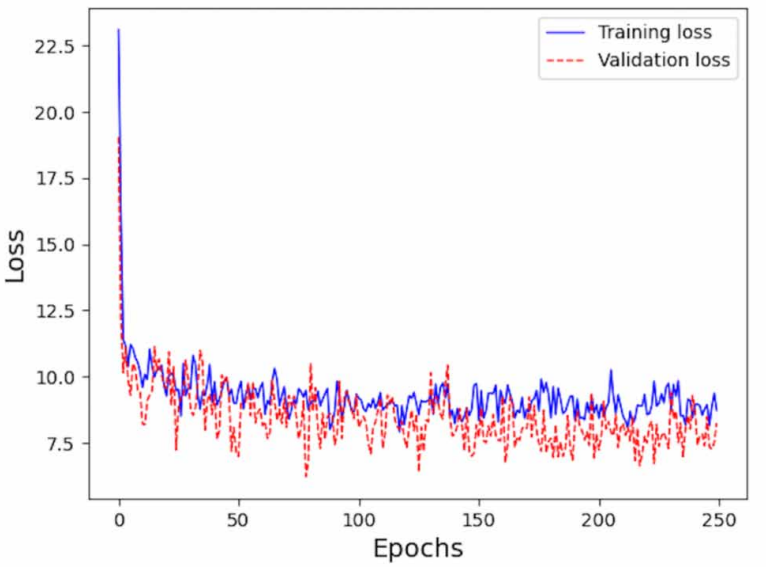


Figure 10. Training and validation loss in one-shot settings using FO-MAML



250, batch size = 64, and optimizer = SGD. The training and validation loss of meta-learner using MAML in the five-shot configuration is shown in Figure 11.

Similarly, we successfully predicted the head pose angles with a MAE_{avg} of 6.84 using FO-MAML. Figure 12 shows the meta-training and meta-validation loss in terms of MAE. Again, the training and validation loss curves show more stability in training MAML than FO-MAML. The MAML algorithm took about $7.44\ \mu\ sec$ for a single outer-loop update whereas, it only took about $5.69\ \mu\ sec$ in FO-MAML.

Ten-Shot Settings Using MAML and FO-MAML

Ten samples from each of five different individuals are randomly chosen for selecting support and query sets for training MAML and FO-MAML in ten-shot scenarios. The model uses mean absolute error (MAE) for performance evaluation in predicting head pose angles. In a ten-shot configuration, the head pose angles have been predicted with a MAE_{avg} of 5.32 using MAML. A similar setup has been used as that of one-shot settings i.e., episode = 250, batch size = 64, and optimizer = SGD to train the model in ten-shot configurations.

Figure 11. Training and validation loss in five-shot settings using MAML

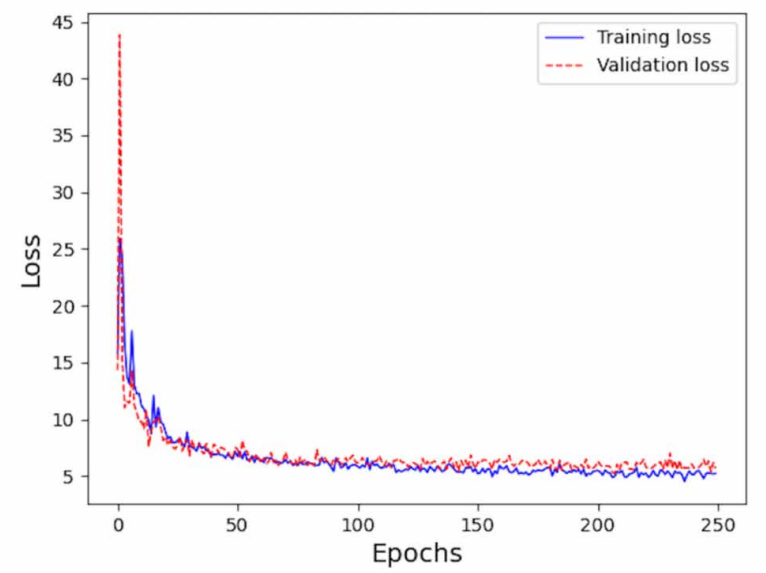
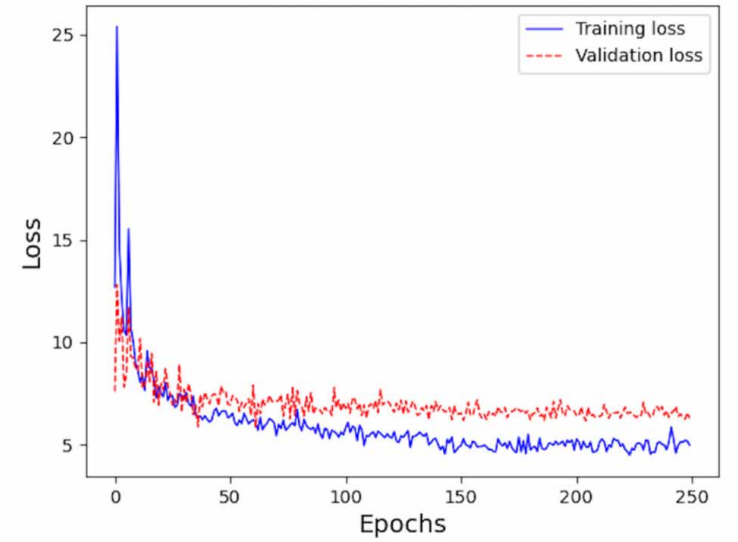


Figure 12. Training and validation loss in five-shot settings using FO-MAML



The training and validation loss of meta-learner using MAML in the ten-shot setting is shown in Figure 13. Similarly, the head pose angles have been predicted with a MAE_{avg} of 6.23 using FO-MAML. Figure 14 shows the meta-training and meta-validation loss in terms of MAE. Again, the training and validation loss curves show more stability in training MAML than FO-MAML. The MAML algorithm took about $9.30\ \mu\ \text{sec}$ for a single outer-loop update whereas, it only took about $6.91\ \mu\text{sec}$ in FO-MAML.

Table 1 summarizes the mean average errors in one-shot, five-shot, and ten-shot settings. The results show slightly better performance using MAML algorithm in few-shot settings. Almost similar

Figure 13. Training and validation loss in ten-shot settings using MAML

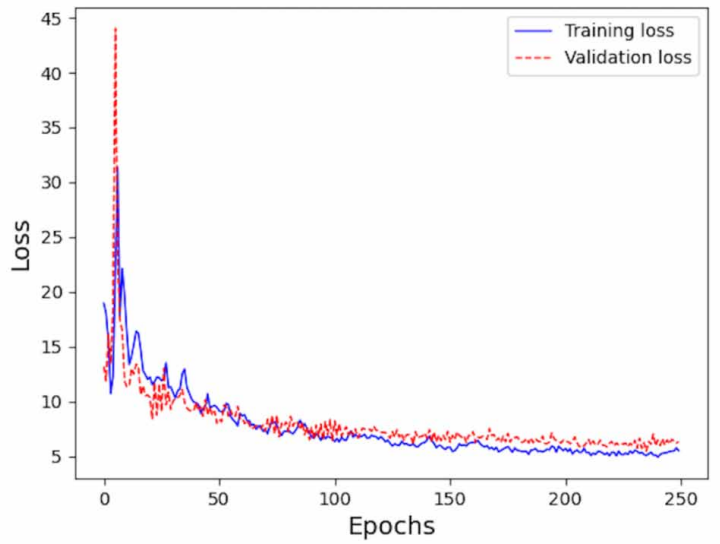
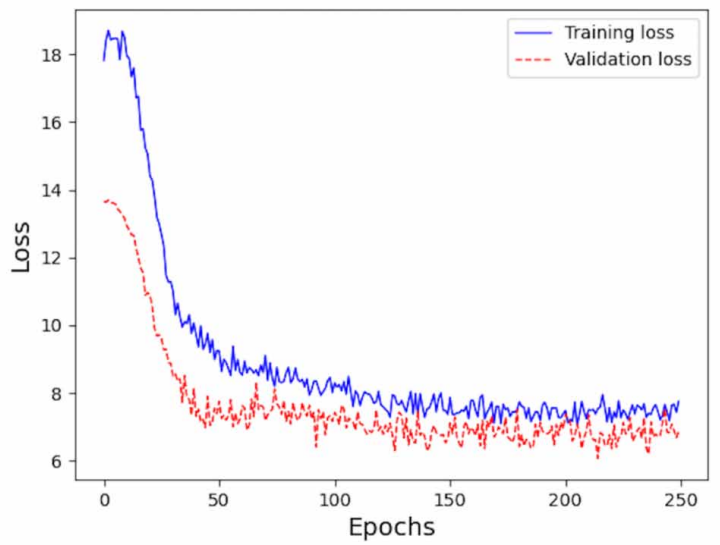


Figure 14. Training and validation loss in ten-shot settings using FO-MAML



performance has been achieved using FO-MAML for estimating head poses. These results confirm that there is almost similar performance while we use FO-MAML instead of MAML. Ignoring second-order gradients doesn't make a huge difference during meta-training. The findings further confirm that the MAML improvement is mostly composed of the objective's gradients after updating the parameter values. The second-order gradients are less significant for the performance of MAML. The experiments show that the FO-MAML algorithm is computationally more efficient in updating gradients than MAML. A single outer-loop update in FO-MAML seems to be much faster than MAML. The computational complexity of an outer-loop update in a single episode in MAML is found to be $O\left(n^2\right)$ whereas, it is linear i.e., $O\left(n\right)$ for FO-MAML.

Table 2 summarizes the comparison of the proposed method with state-of-art on BIWI head pose dataset. For this, we take the results from five-way five-shot settings and make a comparison in computing average MAE.

Comparison with state-of-methods shows that MAML and FO-MAML based approach shows comparable results in estimating head poses. Unlike existing methods, MAML and FO-MAML based methods adapt more faster to unseen tasks that might come in future.

Table 3 summarizes the average time taken to predict head pose in one-shot, five-shot and ten-shot settings. The values are taken by averaging the results of five experiments. The results show FO-MAML is computationally more efficient than MAML in head pose estimation.

Fast Adaptation

A separate unseen test dataset has to be used to analyze the fast adaptation by meta-learner. In this article, we separated a small subset of BIWI dataset containing 2638 faces of five different subject to

Table 1. Mean average error in predicting head pose in one, five and ten-shot settings

Method	5-way 1-shot settings (MAE)	5-way 5-shot settings (MAE)	5-way 10-shot settings (MAE)
Model-Agnostic Meta-Learning (MAML)	**7.72**	**6.30**	**5.32**
First-Order Model-Agnostic Meta-Learning (FO-MAML)	8.33	6.84	6.23

Table 2. Comparison of MAML and FO-MAML based methods to state-of-art head pose estimation methods

Method	Average MAE
Dlib (68 points) (Kazemi & Sullivan, 2014)	12.2
Hopenet (a = 1) (Ruiz, Chong & Rehg, 2018)	4.90
FSA-Caps-Fusion (Yang, Chen, Lin & Chuang, 2019)	4.00
Model-Agnostic Meta-Learning (MAML)	**6.30**
First-Order Model-Agnostic Meta-Learning (FO-MAML)	**6.84**

Table 3. The average time taken to converge by MAML and FO-MAML in one, five and ten-shot settings

Method	5-way 1-shot settings (Avg Time Taken)	5-way 5-shot settings (Avg Time Taken)	5-way 10-shot settings (Avg time taken)
Model-Agnostic Meta-Learning (MAML)	7.15 μs	7.44 μs	9.30 μs
First-Order Model-Agnostic Meta-Learning (FO-MAML)	**5.41 μs**	**5.69 μs**	**6.91 μs**

analyze fast adaptation. The optimal parameters after meta-training are taken as the initial parameters for testing fast adaptation. We just optimize the inner loop for adapting to unseen tasks for analyzing fast adaptation. For this, we consider ten inner steps for training. We randomly select a task from the test distribution and pass it through the meta-learner. Using optimal MAML parameters, mean average error of 8.10, 7.00, and 6.15 has been obtained in predicting head pose using one-shot, five-shot, and ten-shot respectively. Similarly, by using optimal FO-MAML parameters as the initial parameters, mean average error of 8.65, 7.10, and 6.77 has been obtained in estimating head pose for one-shot, five-shot, and ten-shot respectively.

Comparing the Model Performance and Run-Time Performance by Varying the Inner Gradient Steps

For this experiment, we varied inner gradient steps and measured the mean average error in predicting head pose angles. Similarly, we computed the average time taken for the experiment to be complete in seconds. The same hyperparameters while training MAML were used except for the number of inner gradient loop steps: which were taken 5, 10, 15, and 20 respectively. The experiments were run five times each for both MAML and FO-MAML for 100 episodes (epochs); then the mean average error and the average time taken is noted. For simplicity, we conducted the experiment on five-shot settings. The results are shown in Table 4 and Table 5 respectively.

The results from the experiments reveal that FO-MAML wins the performance when it comes to computational time compared to MAML. Similarly, having a larger number of inner gradient steps does not significantly improve the performance of the model. Approximately similar MAE has been obtained by varying the number of inner gradient steps. The error in the prediction of head pose angles is also quite comparable between MAML and FO-MAML.

Figure 15 and Figure 16 show the model performance in terms of mean average error ($\mathrm{MA}E_{avg}$) and average time to be completed respectively.

CONCLUSION

This article proposed a faster approach to training a head pose estimator network using first-order model-agnostic meta-learning (FO-MAML). Experiments using one-shot, five-shot, and ten-shot

Table 4. Mean average error and the average time taken in five-shot settings using MAML

Inner Steps	Method	MAE	Average time taken (in sec)
5	MAML	6.98	565
10	MAML	6.86	692
15	MAML	6.71	832
20	MAML	6.69	974

Table 5. Mean average error and average time taken in five-shot settings using FO-MAML

Inner Steps	Method	MAE	Average time taken (in sec)
5	FO-MAML	7.46	535
10	FO-MAML	7.21	670
15	FO-MAML	7.17	822
20	FO-MAML	7.13	950

Figure 15. Mean average error with varying number of inner gradient steps for 100 episodes

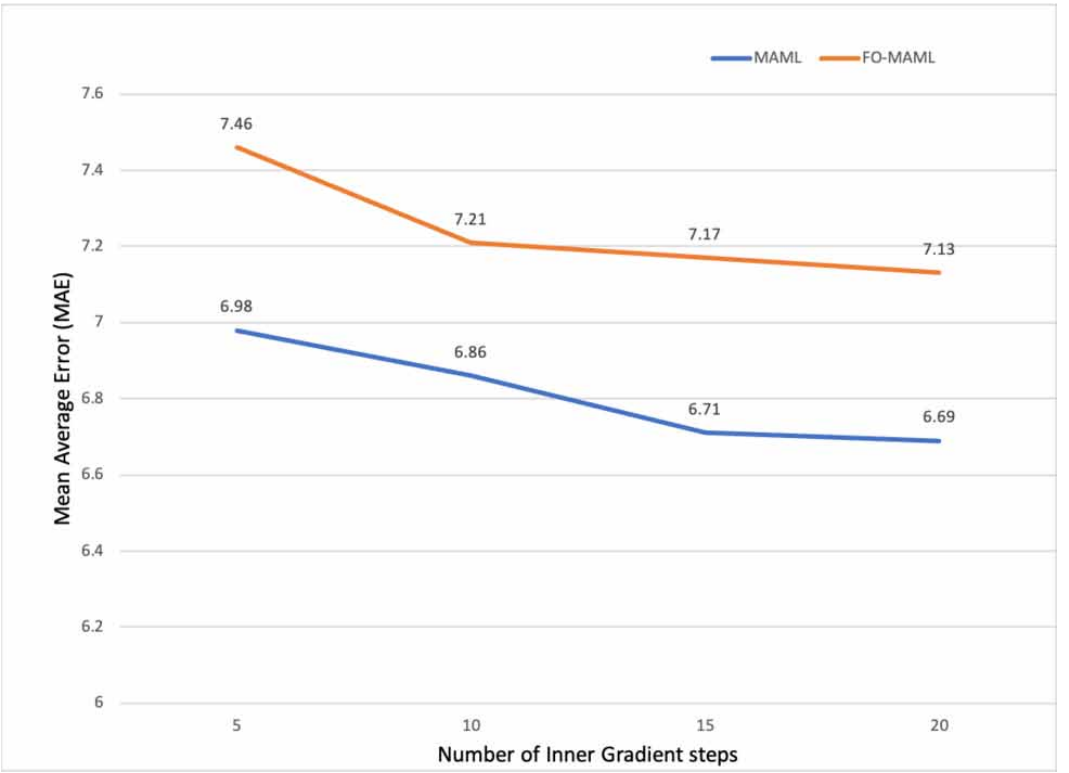
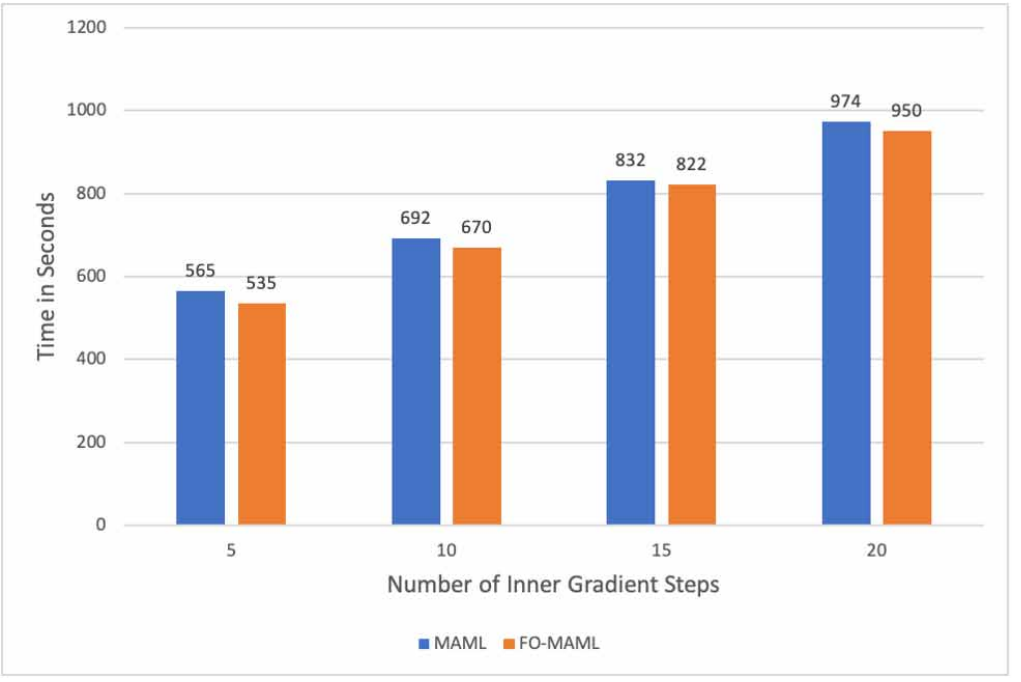


Figure 16. Average time taken with varying number of inner gradient steps for 100 episodes



settings are implemented using MAML and FO-MAML. Mean average error (MAE_{avg}) of 7.72, 6.30, and 5.32 has been achieved using MAML for one-shot, five-shot, and ten-shot settings respectively. Similarly, mean average error (MAE_{avg}) of 8.33, 6.84, and 6.23 is achieved using FO-MAML for one-shot, five-shot, and ten-shot settings. The proposed approach is able to correctly detect head pose angles for faces within the range of ±75∘ in yaw, ±60∘ in pitch, and ±50∘ in the roll using limited training samples. The computational complexity of an outer-loop update in MAML is found to be $\mathrm{O}\left(\mathrm{n}^2\right)$. With FO-MAML the computational complexity of the outer-loop update is found to be linear i.e., $O\left(n\right)$. The results also suggest that second-order gradients are less significant for the performance of MAML. Almost similar performance could be achieved by considering the first-order meta-gradients. The study also confirms that having a larger number of inner gradient steps does not enhance the performance of the model significantly. Almost similar MAE_{avg} has been achieved by varying the number of inner gradient steps. The model has been successful in adapting to the unseen test samples from the BIWI head pose dataset and predicted correct head pose angles using only ten inner-gradient descent steps. The results also support the fact that FO-MAML is computationally more efficient than MAML and gives almost similar performance when used.

The experiments and the results from this approach will definitely give a robust starting ground to create good head pose estimation applications in few-shot settings. However, meta-learning-based approaches have a few challenges when we encounter highly varying task distributions. In the real world, task distribution is often diverse, making it challenging for meta-learners to adapt efficiently. Additionally, for applying meta-learning in few-shot settings, the tasks should be separable to create support and query sets. Many head pose datasets cannot be separated into few-shot tasks, making the application of meta-learning more difficult. In the future, this approach could be verified on more complicated and highly varying datasets. A more efficient approach could be developed by using other meta-learning algorithms such as Reptile and implicit MAML (iMAML).

COMPETING INTERESTS

The authors of this publication declare there are no competing interests.

FUNDING AGENCY

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. Funding for this research was covered by the author(s) of the article.

REFERENCES

Ahn, B., Park, J., & Kweon, I. S. (2015). Real-time head orientation from a monocular camera using deep neural network. In D. Cremers, I. Reid, H. Saito, & M.-H. Yang (Eds.), *Computer Vision – ACCV 2014* (pp. 82–96). Springer International Publishing.

Antoniou, A., Edwards, H., & Storkey, A. (2019). How to train your MAML. *International Conference on Learning Representations*. https://openreview.net/forum?id=HJGven05Y7

Asperti, A., & Trentin, M. (2020). Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 199440–199448. https://doi.org/10.1109/ACCESS.2020.3034828

Ba, S. O., & Odobez, J. (2011). Multiperson Visual Focus of Attention from Head Pose and Meeting Contextual Cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(1), 101–116. doi:10.1109/TPAMI.2010.69 PMID:21088322

Baxter, R., Leach, M., Mukherjee, S., & Robertson, N. (2015). An Adaptive Motion Model for Person Tracking with Instantaneous Head-Pose Features. *IEEE Signal Processing Letters*, *22*(5), 578–582. doi:10.1109/LSP.2014.2364458

Benfold, B., & Reid, I. D. (2008). *Colour Invariant Head Pose Classification in Low Resolution Video*. BMVC.

Bergasa, L., Buenaposada, J., Nuevo, J., Jimenez, P., & Baumela, L. (2008). Analysing Driver's Attention Level using Computer Vision. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 1149 - 1154. doi:10.1109/ITSC.2008.4732544

Fanelli, G., Gall, J., & Van Gool, L. (2011). Real time head pose estimation with random regression forests. *CVPR 2011*, 617–624. 10.1109/CVPR.2011.5995458

Fanelli, G., Weise, T., Gall, J., & Van Gool, L. (2011). Real time head pose estimation from consumer depth cameras. In R. Mester & M. Felsberg (Eds.), *Pattern Recognition* (pp. 101–110). Springer Berlin Heidelberg.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning*, 70, 1126–1135.

Geronimo, D., López, A., Sappa, A., & Graf, T. (2010). Survey of Pedestrian Detection for Advanced Driver Assistance Systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(7), 1239–1258. doi:10.1109/TPAMI.2009.122 PMID:20489227

Grefenstette, E., Amos, B., Yarats, D., Htut, P. M., Molchanov, A., Meier, F., Kiela, D., Cho, K., & Chintala, S. (2019). *Generalized inner loop meta-learning.* ArXiv Preprint ArXiv:1910.01727.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. 10.1109/CVPR.2016.90

Jin, X., & Tan, X. (2016). Face alignment by robust discriminative hough voting. *Pattern Recognition*, *60*, 318–333. Advance online publication. doi:10.1016/j.patcog.2016.05.017

Joshi, M., Pant, D., Karn, R., Heikkonen, J., & Kanth, R. (2022). Meta-Learning, Fast Adaptation, and Latent Representation for Head Pose Estimation. *Proceedings of the XXth Conference of Open Innovations Association FRUCT*, 31. doi:10.23919/FRUCT54823.2022.9770932

Joshi, M., Pant, D. R., & Acharya, B. (2021). Head Pose Estimation by Few Shot Learning Techniques. *Proceedings of 10th IOE Graduate Conference*, *10*, 1391–1397.

Kashevnik, A., Ali, A., Lashkov, I., & Zubok, D. (2021). Human head angle detection based on image analysis. In K. Arai, S. Kapoor, & R. Bhatia (Eds.), Proceedings of the Future Technologies Conference (FTC) 2020 (Vol. 1, pp. 233–242). Springer International Publishing. https://doi.org/10.1007/978-3-030-63128-4_18.

Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1867–1874. 10.1109/CVPR.2014.241

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324. https://doi.org/10.1109/5.726791

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., & Stoica, I. (2018). *Tune: A Research Platform for Distributed Model Selection and Training.* ArXiv, abs/1807.05118.

Liu, Q., Yang, J., Deng, J., & Zhang, K. (2016). Robust facial landmark tracking via cascade regression. *Pattern Recognition*, *66*, 53–62. Advance online publication. doi:10.1016/j.patcog.2016.12.024

Ng, J., & Gong, S. (1999). Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. *Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No.PR00378)*, 14–21. doi:10.1109/RATFG.1999.799218

Osadchy, M., Miller, M., & Lecun, Y. (2004). Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research, 8*. 10.1007/11957959

Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: An application to face detection. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 130–136. doi:10.1109/CVPR.1997.609310

Park, S., Mello, S., Molchanov, P., Iqbal, U., Hilliges, O., & Kautz, J. (2019). *Few-shot adaptive gaze estimation*. 10.1109/ICCV.2019.00946

Patacchiola, M., & Cangelosi, A. (2017). Head pose estimation in the wild using Convolutional Neural Networks and adaptive gradient methods. *Pattern Recognition*, *71*, 132–143. doi:10.1016/j.patcog.2017.06.009

Patacchiola, M., & Cangelosi, A. (2017). Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition, 71*. 10.1016/j.patcog.2017.06.009

Ruiz, N., Chong, E., & Rehg, J. M. (2018). Fine-grained head pose estimation without keypoints. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2155–215509. 10.1109/CVPRW.2018.00281

Sherrah, J., Gong, S., & Ong, E. (1999). *Understanding Pose Discrimination in Similarity Space*. BMVC. doi:10.5244/C.13.52

Srinivasan, S., & Boyer, K. L. (2002). Head pose estimation using view based eigenspaces. *2002 International Conference on Pattern Recognition*, *4*, 302–305. doi:10.1109/ICPR.2002.1047456

Sun, Q., Liu, Y., Chua, T.-S., & Schiele, B. (2019). *Meta-transfer learning for few-shot learning*. 10.1109/CVPR.2019.00049

Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3476–3483. doi:10.1109/CVPR.2013.446

Sun, Q., Liu, Y., Chua, T.-S., & Schiele, B. (2018). Meta-transfer learning for few-shot learning. *arXiv*. 10.48550/ARXIV.1812.02391

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). *Going deeper with convolutions*. arXiv. 10.48550/ARXIV.1409.4842

Tran, D. T., & Lee, J.-H. (2011). A Robust Method for Head Orientation Estimation Using Histogram of Oriented Gradients. *Communications in Computer and Information Science*, *260*, 391–400. doi:10.1007/978-3-642-27183-0_41

Venturelli, M., Borghi, G., Vezzani, R., & Cucchiara, R. (2017). *Deep head pose estimation from depth data for in-car automotive applications*. arXiv. 10.48550/ARXIV.1703.01883

Yan, Y., Ricci, E., Subramanian, R., Liu, G., Lanz, O., & Sebe, N. (2016). A multi-task learning framework for head pose estimation under target motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(6), 1070–1083. https://doi.org/10.1109/TPAMI.2015.2477843

Yang, T.-Y., Chen, Y.-T., Lin, Y.-Y., & Chuang, Y.-Y. (2019). Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1087–1096. 10.1109/CVPR.2019.00118

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

Zhu, Y., & Fujimura, K. (2004). Head pose estimation for driver monitoring. *IEEE Intelligent Vehicles Symposium*, 501–506. doi:10.1109/IVS.2004.1336434