

# ETL: Efficient Transfer Learning for Face Tasks

Thrupthi Ann John<sup>1</sup>, Isha Dua<sup>1</sup>, Vineeth N Balasubramanian<sup>2</sup> and C. V. Jawahar<sup>1</sup>

<sup>1</sup>Center for Visual Information Technology, International Institute of Information Technology, Hyderabad, India

<sup>2</sup>Department of Computer Science and Engineering, Indian Institute of Technology, Hyderabad, India  
{thrupthi.ann, isha.dua}@research.iit.ac.in, vineethnb@iith.ac.in, jawahar@iit.ac.in

**Keywords:** Face Tasks, Transfer Learning, Efficient Transfer Learning, Face Recognition, Expression Recognition, Age Prediction, Gender Prediction, Head Pose

**Abstract:** Transfer learning is a popular method for obtaining deep trained models for data-scarce face tasks such as head pose and emotion. However, current transfer learning methods are inefficient and time-consuming as they do not fully account for the relationships between related tasks. Moreover, the transferred model is large and computationally expensive. As an alternative, we propose ETL: a technique that efficiently transfers a pre-trained model to a new task by retaining only *cross-task aware filters*, resulting in a sparse transferred model. We demonstrate the effectiveness of ETL by transferring VGGFace, a popular face recognition model to four diverse face tasks. Our experiments show that we attain a size reduction up to 97% and an inference time reduction up to 94% while retaining 99.5% of the baseline transfer learning accuracy.

## 1 INTRODUCTION

Deep neural networks are widespread in machine learning, achieving state-of-the-art results in most face-related tasks. However, they are known to be highly data and compute-hungry. Massive face datasets with millions of images, such as VGG-Face2 (Cao et al., 2018) which contains 3M images, or Ms-Celeb-1M (Guo et al., 2016) which has 10M images partially solve the first problem. While large datasets exist for face tasks such as recognition, other tasks such as age or emotion recognition have comparatively very little publicly available data due to the difficulty of collecting and annotating data. Thus, transfer learning is popular, where we take a model trained on a ‘primary task’ with lots of data and transfer it to a secondary task using finetuning. However, the resulting model is still large and computationally intensive, and the transfer learning process is time-consuming and does not fully utilize the learned filter information from the primary model.

Previously, many papers (Oquab et al., 2014; Razavian et al., 2014) have shown the generalization capability of deep convolution network across various tasks. This is possible because tasks are often related, and when a deep neural network learns to predict a given task, the feature representation it learns can be

adapted to other similar tasks to varying degrees. Several efforts in recent years (Donahue et al., 2014; Khorrani et al., 2015; Long et al., 2014; Zhou et al., 2014) have found such relationships between tasks that are diverse but related, such as object detection to image correspondence (Long et al., 2014), scene detection to object detection (Zhou et al., 2014) and expression recognition to facial action units (Khorrani et al., 2015). Similarly, it is no new fact that tasks in the face domain are highly related to each other. As much as face tasks have to deal with many variations in images, different face tasks (such as face recognition, pose estimation, age estimation, emotion detection) operate on input data that are fairly similar to each other (John et al., 2021). These face tasks attempt to capture fine-grained differences between the images. Since the tasks are related and come from the same domain, learning one task can help learn other tasks.

To this end, we propose ETL: an efficient transfer learning method for faces that is based on understanding the impact of different filters in a convolutional layer of a primary model with respect to the secondary tasks for which the model is not trained. Figure 1 illustrates our method. We identify convolutional filters from the primary model that are not relevant to the secondary task using lasso regression and remove them in a one-pass pruning step. The resulting sparse model is then fine-tuned for the respective secondary

<sup>a</sup><https://orcid.org/0000-0002-8557-6564>

<sup>b</sup><https://orcid.org/0000-0001-5494-059X>

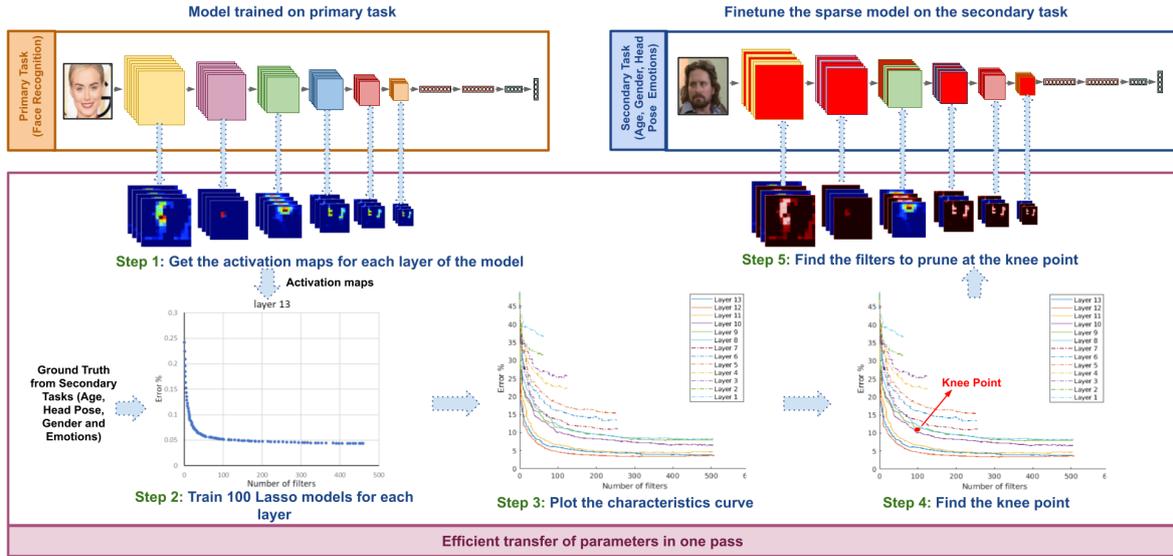


Figure 1: Pipeline for efficient transfer of parameters from model trained on primary task like face-recognition to model for secondary task including gender, emotion, head pose and age in one pass. The ETL technique identifies and preserves the task related filters only which in turn results in highly sparse network for efficient training of face related tasks.

task. Our method significantly reduces the training time as compared to training from scratch or standard transfer learning and produces computationally light models without compromising on performance. The proposed approach has application in various domains including ADAS (Dua et al., 2019; Dua et al., 2020b; Dua et al., 2020a) which requires efficient implementation of face algorithms in real time applications. The proposed transfer learning technique has the following advantages:

1. **Rapid transfer learning** Our approach is non-iterative, as we identify all non-relevant filters in a single pass using lasso regression, unlike other pruning methods which iteratively prune filters and fine-tune the model.
2. **Light-weight models** Our approach achieves high compression-ratio, which results in faster training times and real-time inference times without compromising on accuracy, which is important for deployment to low-powered edge devices.
3. **Requires less data** ETL leverages existing filters from primary models to train models on tasks with less available data.

We conduct extensive experiments to validate our proposed approach and compare it to the standard transfer learning algorithm. We present our results on multiple face datasets, covering secondary tasks like age, gender, emotions and head pose, for which large datasets do not exist.

## 2 RELATED WORK

**Transfer Learning:** In traditional transfer learning (Bengio, 2012; Bengio et al., 2011; Caruana, 1995; Aytar and Zisserman, 2011; Lim et al., 2011; Oquab et al., 2014; Tommasi et al., 2010), a model trained on a base task is finetuned on a target data set/task. Several exploratory studies have investigated best policies and practices for transfer learning by conducting large-scale experiments on various tasks. (Zamir et al., 2018) use a computational approach to recommend the best transfer learning policy between a set of source and target tasks. They also find structural relationships between vision tasks using this approach. (Yosinski et al., 2014) provide many recommendations for best practices in transfer learning. They quantify the degree to which a particular layer is general or specific, i.e., how well features at that layer transfer from one task to another. They also quantify the ‘distance’ between different tasks using a computational approach.

**Lightweight convolution models:** Current deep learning models show impressive performance at the cost of having a lot of parameters, which makes them energy-inefficient and challenging to deploy on low-end devices. To date, several studies have investigated various architectures for lightweight convolution models for faster training with minimal loss in performance. (Szegedy et al., 2015) proposed inception modules which decrease the channels to expensive 3x3 convolutions. (Chollet, 2017) and (Howard

et al., 2017) took this further to make 3x3 convolutions completely depthwise separable and sparse. (Iandola et al., 2016) further reduced parameters by downsampling late in the network so that convolution layers have large activation. (Hitawala, 2018), (Zhang et al., 2018) and (Wu et al., 2018) employed grouped convolutions to get efficient models. Recently, (Duong et al., 2019b) and (Sharma and Foroosh, 2020) proposed lightweight CNN architectures designed for face tasks. An alternative to specially designed CNN architectures is quantized networks (Hubara et al., 2017; Gong et al., 2014; Kim and Smaragdis, 2016; Rastegari et al., 2016; Miyashita et al., 2016) which are neural networks with extremely low precision. They replace most arithmetic operations with bitwise operations and drastically reduce memory and power consumption.

Another strategy is to start with a massive network and reduce its size using pruning or knowledge distillation. Pruning involves removing connections from a complete network based on some ranking criterion to obtain a sparse network with similar performance as the initial network. Connections may be pruned at different resolutions, such as at the neuron or filter level. Recent research (Li et al., 2016; Luo et al., 2018; He et al., 2018) explored various criteria for ranking convolutional filters and removed the bottom k% of the filters iteratively. A notable work is (He et al., 2017), which selects filters by a lasso regression-based method and least-square reconstruction in an iterative manner. In contrast, we use lasso regression to select filters in one pass. Some works (Lee et al., 2019; Zhang and Stadie, 2020) pruned connections in one shot, but they operated on the neuron resolution. Recently, various works approached pruning using the 'Lottery Ticket Hypothesis' (Frankle and Carbin, 2019) which naturally uncovers subnetworks whose initialization made them capable of training effectively.

On the other hand, knowledge distillation starts with a large trained 'teacher' model and transfers the knowledge to a smaller 'student' model. The student model is trained on the output distribution of the teacher model instead of the ground truth labels. Several works (Jin et al., 2019; Antipov et al., 2017; Duong et al., 2019a) achieved impressive performance with lightweight models using knowledge distillation on face tasks such as recognition, detection and age estimation.

**Efficient transfer learning:** While these approaches solve storage inefficiency, computational complexity, and power consumption problems, they are not designed for task transfer. Recent works in

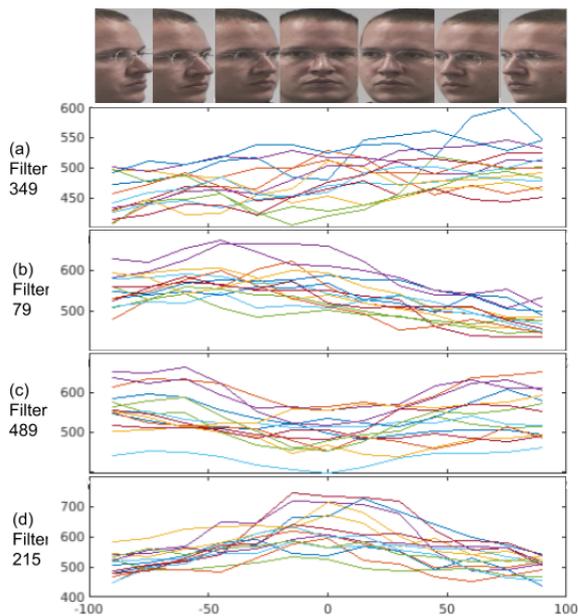


Figure 2: The figure shows the correlation between yaw angle on Head Pose Image Database and average responses of a few convolutional filters from the last layer of VGG-Face. The different lines in each graph represent 15 different identities: (a) high activation for left-facing faces; (b) high response for faces facing right; (c) high response for sideways faces; (d) high response for frontal faces

NLP (Houlsby et al., 2019; Guo et al., 2021; Zhang et al., 2020) focused on efficient incremental learning, where a few additional neurons per task ensures that catastrophic forgetting does not occur and the resulting efficient model achieves the performance of separate complete networks for new tasks. (Wang and Lan, 2017) uses knowledge distillation to transfer from face recognition to non-classification tasks of alignment and verification by choosing the appropriate initializations and targets. (Molchanov et al., 2016) is a closely related work to ours which performs pruning and transfer learning at the same time. They alternate between finetuning and pruning until the required objective of accuracy versus compression is reached. They finetune all model parameters initially, and their approach is iterative and slow, unlike ETL, which transfers the model in one shot.

### 3 METHODOLOGY

The current practice to obtain face models for a data-scarce task is to finetune all the filters of a pre-trained model for the task. However, this method is inefficient and resource-hungry. Our method ETL relies on groups of *Cross-Task Aware Filters* which form a small percentage of all the model filters. ETL identifies the optimum filter sets and finetunes them on the

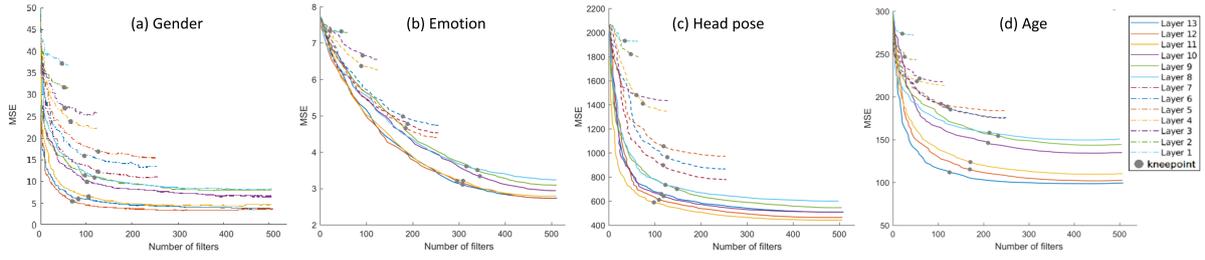


Figure 3: Characteristic curves for all filters of the VGG-Face pre-trained model regressed on gender, emotion, head pose and age. The gray dot shows the knee point. We observe that regression gives a tiny error using as few as  $\sim 100$  filters. Adding more filters to the regression model does not significantly impact the error, indicating that the additional filters do not capture much information about gender.

---

**Algorithm 1:** Create sparse model by removing selected filters

---

**Input:** Model  $\phi$  with  $L$  convolutional layers having weights  $\{W_1^\phi, W_2^\phi, \dots, W_L^\phi\}$ , regression weights  $\{\beta_1, \beta_2, \dots, \beta_L\}$  of knee-point LASSO models for each layer

**Output:** Sparse model  $\phi'$

```

1  $\phi' \leftarrow$  copy of  $\phi$  with weights  $\{W_1^{\phi'}, W_2^{\phi'}, \dots, W_L^{\phi'}\}$ 
2 for each convolutional layer  $l$  of  $\phi$  do
3    $n_l \leftarrow$  number of non-zero elements of  $\beta_l$ 
4   for each non-zero element  $i$  in  $\beta_l$  and  $j=1$  to  $n_l$  do
5      $W_l^{\phi'}[j, :, :, :] \leftarrow \beta_l[i] W_l^\phi[i, :, :, :]$ 
6     if  $l < L$  then
7        $W_{l+1}^{\phi'}[:, j, :, :] \leftarrow W_{l+1}^\phi[:, i, :, :]$ 
8     else
9       Let  $W_{L+1}^\phi \in \mathbb{R}^{C_{L+1} \times C_{L+2}}$  be the first linear layer of model  $\phi$ 
10       $W_{L+1}^{\phi'}[j, :] \leftarrow W_{L+1}^\phi[i, :]$ 
11    end
12  end
13 end

```

---

new task. The rest of the filters are discarded before finetuning, which results in a compact model with reduced training time. In the sections below, we show the existence of such filters and discuss our proposed procedure for Efficient Transfer Learning.

### 3.1 Motivation

Do models trained on a face task like recognition contain information about other related face tasks? Experiments show that some convolutional filters (channels of a convolutional layer) of face recognition models learn to predict related face tasks such as

head pose, age and gender without additional supervision. We call these filters *Cross-task Aware Filters* (CRAFTs). We demonstrate the presence of CRAFTs with the following experiment. The VGG-Face model (Parkhi et al., 2015) is trained for face recognition on 2.6 million images. We find CRAFTs for head pose in its final convolutional layer using the Head Pose Image Database (Gourier et al., 2004), which has face images with all attributes kept constant except head pose. We pass the dataset images through the model and plot the mean of each final layer filter activation map against the yaw of the head. Figure 2 shows some highly correlated filter activations w.r.t yaw. Some filters give a high response to front-facing images, whereas others respond strongly to face images turned to one side. These CRAFTs formed in the face recognition model without additional supervision or explicit training for yaw. We can use them for transferring the model to predict the yaw of the head.

The following sections discuss how we find the optimal CRAFT sets for different secondary tasks like Age, Head Pose, Gender and Emotions and use them for efficient transfer learning.

### 3.2 Finding optimal sets of CRAFTs

We now find the optimal CRAFT sets which predict secondary tasks like Age, Head Pose, Gender and Emotion. Let  $D = (I, Y)$  be a dataset for a secondary task where  $I \in \mathbb{R}^{N \times 3 \times W \times H}$  is the set of  $N$  dataset images and  $Y \in \mathbb{R}^N$  is the corresponding ground-truth values. Consider the  $l^{\text{th}}$  convolutional layer of a model  $\phi$  having weights  $W \in \mathbb{R}^{C_{l+1} \times C_l \times k \times k}$ , which has  $C_{l+1}$  output channels/filters. Let  $\phi_l(I) \in \mathbb{R}^{N \times C_{l+1} \times w \times h}$  be the activation of layer  $l$ . Let  $X \in \mathbb{R}^{N \times C}$  be the average activations:

$$X = \frac{1}{wh} \sum_{i=1}^w \sum_{j=1}^h \phi_l(I)[:, :, i, j] \quad (1)$$

We need to choose groups of filters whose activations are highly correlated with  $Y$ . One way to do this is to rank each filter group based on a correlation coefficient  $\rho$  and pick the highest-ranked filters.

$$\rho_c = \left| \frac{\text{Cov}(X[:,c], Y)}{\sigma_{X[:,c]} \sigma_Y} \right| \quad (2)$$

where  $X[:,c] \in \mathbb{R}^N$  is the activation of the  $c^{\text{th}}$  filter. However, individually picking filters results in a greedy solution as we do not consider the interdependence of filters. Instead of exhaustively checking all groups of filters in a layer, we use LASSO (Tibshirani, 1996), an  $L_1$ -regularized regression method which selects a subset of filters that best predict  $Y$  using the objective:

$$\min_{\beta_0, \beta} \left( \frac{1}{2N} \sum_{i=1}^N (Y_i - \beta_0 - X_i^T \beta_i)^2 + \lambda \sum_{j=1}^p |\beta_j| \right) \quad (3)$$

where  $Y_i$  is the ground truth of sample  $i$ ,  $X_i \in \mathbb{R}^{C_{l+1}}$  is the global average-pooled activation of sample  $i$ ,  $\beta \in \mathbb{R}^{C_{l+1}}$  is the LASSO regression weight vector and  $\lambda$  is a non-negative regularization parameter which determines the sparseness of the regression weights  $\beta$ . The number of filters chosen decreases with an increase in  $\lambda$ , as more coefficients of  $\beta$  become zero.

### 3.3 Characteristic curves

Choosing a different  $\lambda$  for each layer is non-trivial as a change in  $\lambda$  does not cause a corresponding change in error. In this section, we define a global hyperparameter that balances the trade-off between sparsity and error, eliminating the need for per-layer sensitivity parameters.

To see how error varies with sparsity, we create *characteristic curves* for each layer, which is a plot of the sparsity of filters versus the error for different values of  $\lambda$ . We train 100 LASSO models by varying the  $\lambda$  such that the largest  $\lambda$  just makes all the coefficients zero. The rest of the  $\lambda$  values are chosen using a geometric sequence such that the ratio of largest to smallest  $\lambda$  is  $1e+4$ . Figure 3 shows some examples of characteristic curves for various secondary tasks. We notice that the characteristic curves are flat-bottomed for some tasks, i.e. there is no significant change in error as the sparsity increases. The ‘shape’ of the characteristic curves vary with the secondary task.

We define a global parameter  $\gamma$ , which is the maximum allowed increase in error. We define the knee point of the curve  $k$  as the  $\lambda$  value that maximizes the sparsity while keeping the error within limits.

$$k = \min_i \text{num}(i) \text{ such that } \text{RMSE}(i) - \min(r) < \gamma(\max(r) - \min(r)) \quad (4)$$

where  $i$  is the  $\lambda$  value at a point on the characteristic curve,  $\text{num}(i)$  is the number of filters chosen when  $\lambda = i$ ,  $\text{RMSE}(i)$  is the RMS error of the LASSO model with  $\lambda = i$  and  $\min(r)$  and  $\max(r)$  are the minimum and maximum RMS error values for all the LASSO models in the curve respectively. A higher  $\gamma$  indicates that the transferred model will be larger with lower error and vice versa. We calculate the  $\lambda$  value at knee-point  $k$  for each layer using the chosen  $\gamma$  parameter.

### 3.4 Obtaining a sparse model

In this step, we discard all the filters not chosen by the LASSO model with  $\lambda = k$  for each convolutional layer of the model. We follow the procedure in (Li et al., 2017). Consider the  $l^{\text{th}}$  convolutional layer of the model  $\phi$  whose kernel size is  $k_l \times k_l$ . Its weight matrix  $W$  is of size  $C_{l+1} \times C_l \times k_l \times k_l$  where  $C_l$  refers to the input channels of layer  $l$  and  $C_{l+1}$  is the number of output channels of layer  $l$ , or the input channels of layer  $l+1$ . The weight matrix of the  $l+1^{\text{th}}$  layer is  $W_{l+1} \in \mathbb{R}^{C_{l+2} \times C_{l+1} \times k_{l+1} \times k_{l+1}}$ . In order to remove the  $i^{\text{th}}$  filter from layer  $l$ , the output channel weight  $W_l[i, :, :, :]$  is removed. The corresponding input channel weight  $W_{l+1}[:, i, :, :]$  is removed from layer  $l+1$ . We remove filters from all the layers in one shot according to the LASSO model at the chosen knee point  $k$ . Let  $\beta$  be the LASSO regression weight vector and  $t \in 1..C_{l+1}$  be the index of the filters chosen when  $\lambda = k$ , which are the non-zero coefficients of  $\beta$ . The new weight vector of layer  $l$  is given by

$$W'_l = \beta[t] W_l[t, :, :, :] \quad (5)$$

The detailed algorithm is given in Algorithm 1.

### 3.5 Efficient transfer learning

Our complete pipeline is given in Figure 1. We begin with an initial model  $\phi$  pre-trained on a primary task  $D_1$ . Let  $D_2 = (I, Y)$  be the secondary task. We first pass the dataset images  $I$  through the model  $\phi$  and collect the activations at each layer  $\{X_1, X_2, \dots, X_L\}$  according to Equation 1. We then plot the characteristics curve and find the knee-point  $k_l$  for each layer using Equation 4. We generate a sparse model  $\phi'$  by keeping only filters corresponding to the non-zero coefficients of the regression weights  $\beta_l$  of the LASSO models with  $\lambda = k_l$ , according to Algorithm 1. Finally, we finetune the sparse model  $\phi'$  on the dataset  $D_2$  to obtain the efficient transferred model  $\phi^*$ .

Table 1: The table shows the comparison of ETL with Transfer Learning in terms of accuracy, FLOPS, size, and inference time per image on CPU for different face tasks, including gender, emotion, head pose, and age. The percentage reduction in metrics is given in the brackets. We observe a significant drop in model size, which leads to faster inference time with a slight loss in the model’s accuracy.

		<b>Gender</b>	<b>Emotion</b>	<b>Head pose</b>	<b>Age</b>
<b>Baseline transfer learning</b> (with full fine tuning)	<b>Accuracy</b>	<b>97.06</b>	<b>65.92</b>	<b>95.7</b>	<b>51.8</b>
	<b>FLOPS</b>	7.38E+11	7.38E+11	7.38E+11	7.38E+11
	<b>Size</b>	5.80E+08	5.80E+08	5.80E+08	5.80E+08
	<b>Inference</b>	5.469	5.527	5.5169	5.313
<b>ETL (Our method with sparse fine tuning)</b>	<b>Accuracy</b>	96.62 (0.5%)	55.16 (16.3%)	94.58 (1.2%)	46.96 (9.3%)
	<b>FLOPS</b>	<b>3.26E+11</b> (55.8%)	<b>2.06E+11</b> (72.1%)	<b>4.25E+11</b> (42.4%)	<b>4.16E+11</b> (43.6%)
	<b>Size</b>	<b>2.59E+07</b> (95.5%)	<b>1.64E+07</b> (97.2%)	<b>3.39E+07</b> (94.2%)	<b>3.32E+07</b> (94.3%)
	<b>Inference</b>	<b>0.528</b> (90.3%)	<b>0.373</b> (94.2%)	<b>0.4626</b> (91.6%)	<b>0.47</b> (91.2%)

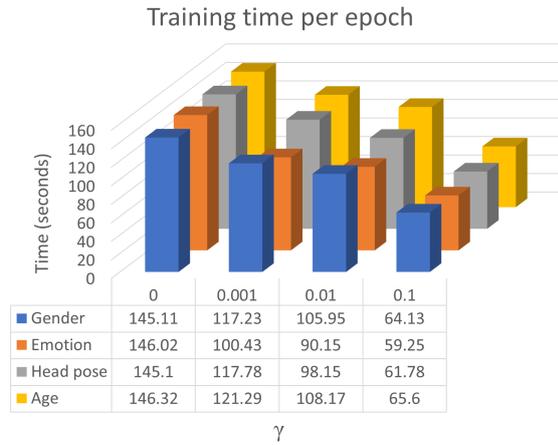


Figure 4: Training time per epoch on GPU for threshold values between 0 to 0.1

## 4 EXPERIMENTS AND RESULTS

This section shows that ETL achieves fast and parameter efficient transfer learning for face tasks when compared to the baseline transfer learning method. Experiments on several face datasets show that the ETL models retains up to 99.5% of the baseline accuracy while reducing the size of the baseline model by 97%, thereby reducing the CPU inference time by 94%.

### 4.1 Evaluation metrics

For our experiments, we measure the accuracy, the FLOPs for a forward pass, the number of parameters of the model, and the inference time as the criteria to compare our methods. *FLOPS*: We calculate FLOPs as the number of multiplication operations required for a forward pass. For a model with  $M$  convolutional layers and  $N$  linear layers, we calculate the FLOPs as follows:

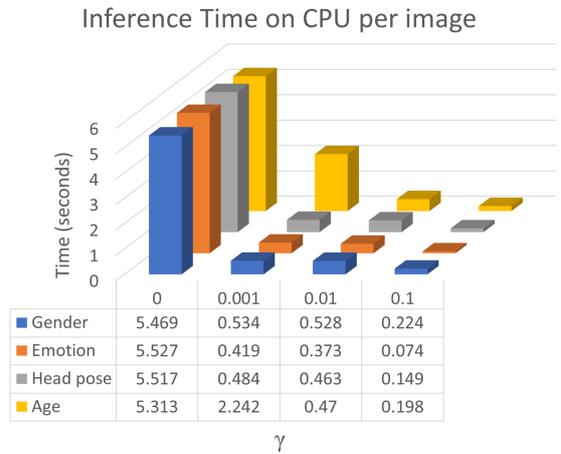


Figure 5: Inference time on CPU per image at different threshold values. The increase in threshold value results in higher real-time performance.

$$FLOPS = \sum_{i=1}^M ConvFLOPS_i + \sum_{j=1}^N LinFLOPS_j \quad (6)$$

$$ConvFLOPS_l = o_l \times i_l \times k_l \times k_l \times w_l \times h_l \quad (7)$$

$$LinFLOPS_l = n_{input_l} \times n_{output_l} \quad (8)$$

Here,  $o_l$  and  $i_l$  refer to the number of output and input channels, the kernel is of size  $k_l \times k_l$  and the activation map is of size  $o + l \times w_l \times h_l$  of convolutional layer  $l$ ; and  $n_{input_l}$  and  $n_{output_l}$  are the number of input and output features for linear layer  $l$ .

*Size of the network* is the sum of the sizes of its stored parameters, consisting of various layers’ weights and biases. It affects the resource and time required for training the deep models. *Inference Time* is the time required to predict the output of one image at test time on a CPU. A low inference time signifies the possibility of using the deep model on devices with restricted resources.

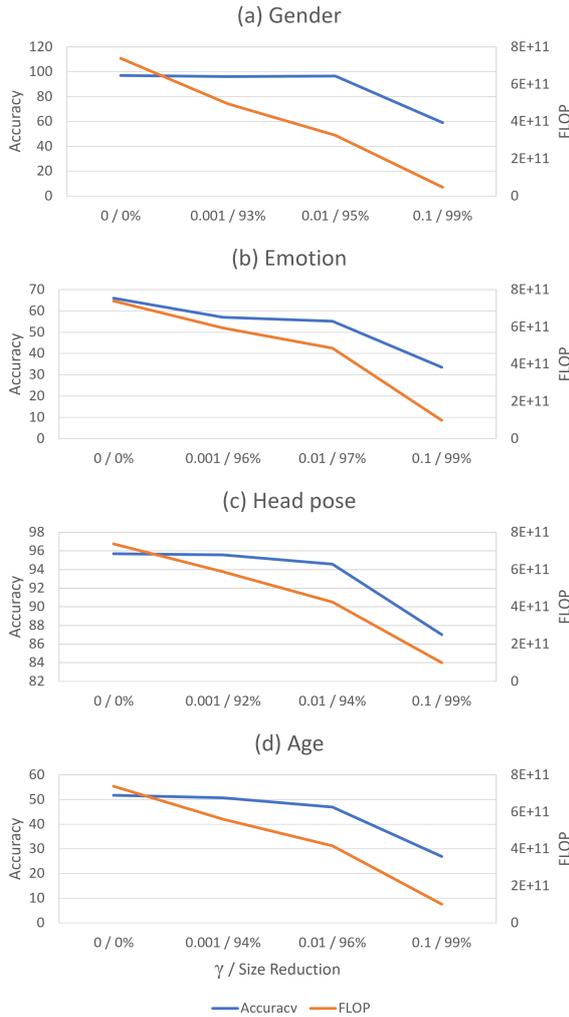


Figure 6: The four figures show the accuracy and computational complexity for the VGG-Face model pruned with different thresholds( $\gamma$ ). For each task, we varied the threshold from 0.1 to 0.001. A threshold of 0 indicates an unpruned network, and a threshold of 0.1 corresponds to a highly sparse network with 99% of filters pruned. We have shown the accuracy on the left axis and computational cost (number of flops) on the right axis. The X-axis shows the percentage reduction in size along with the respective threshold values on the X-axis. The four figures correspond to the different face tasks: a) Gender b) Emotion c) Head pose d) Age.

## 4.2 Experimental Setup

We use the public, pre-trained VGGFace (Parkhi et al., 2015) model for face recognition as our base model. Using our efficient transfer learning procedure, we transfer the VGGFace model to the four tasks mentioned above of gender, emotion, head pose, and age. We compare the results obtained using the proposed technique (ETL) with the baseline transfer

learning technique, where all the filters are finetuned for the new task.

We conduct experiments on four different face datasets. Annotated Facial Landmarks in the Wild (AFLW) (Martin Koestinger and Bischof, 2011) is a large-scale dataset of face images in the wild annotated with head pose and landmarks. We use the ‘yaw’ component of the head pose expressed in radians for our task. AgeDB is a collection of face images annotated with the age of the person. The values range from 0 to 101. The AFEW-VA database for valence and arousal estimation in-the-wild (Kossaifi et al., 2017; A. Dhall and Gedeon, 2012) is a collection of per-frame annotations of valence and arousal for 600 challenging video clips extracted from feature films. We treat this dataset as a collection of images (without the temporal component) and use ‘valence’ labels as our task. The CelebA dataset (Liu et al., 2015) consists of 202,599 face images for which the ground truth values of 40 attributes are provided. We use the attribute ‘gender’ in our experiments. The data sets are randomly split into 75% for training and 25% for testing.

## 4.3 Results

We compare our proposed ETL procedure with baseline transfer learning for the tasks of gender, emotion, head pose and age. Table 1 summarizes the results of ETL with  $\gamma = 0.01$ . We observe a significant reduction of up to 97% in size and 72% in the computational complexity without much loss of accuracy, as we can remove many convolutional filters from each layer without impacting the performance. We observe from Figure 3 that the characteristics curves for gender and head pose are flat, indicating that most of the information about secondary tasks exists in very few filters of each convolutional layer of the VGG-Face network. Thus, the performance of the ETL models reaches up to 99.9% of the baseline models. The characteristics curves for emotion and age are not as flat, resulting in a higher performance drop.

The value of  $\gamma$  controls the model compactness; higher  $\gamma$  results in fewer parameters at a possible cost to the performance. To explore this trade-off, we consider different  $\gamma$  values and compare their effect on accuracy, FLOPs and size to the baseline. Figure 6 presents our results. Using VGGFace as the base network, we applied our ETL procedure for four tasks: gender, emotion, head pose and age with  $\gamma$  values of 0, 0.1, 0.01 and 0.001. The figure shows that the FLOPs reduce monotonically as  $\gamma$  changes. We observe that as  $\gamma$  increases, the model size and computational complexity reduces significantly with only a minor reduc-

tion of accuracy. Thus, the threshold is a reliable way to tune the ETL algorithm and get the desired compromise between compression ratio and accuracy. In our experiments, we observed a  $\gamma$  value of 0.01 as ideal.

Figure 4 shows the training time per epoch for different values of  $\gamma$ , which reduces with an increase in  $\gamma$  as fewer filters get chosen. We observe a per-epoch reduction of 32% for  $\gamma = 0.01$  for head pose. This speeds up the finetuning step, resulting in accelerated transfer learning. Figure 5 presents the inference time on CPU per image at different  $\gamma$  values. A dramatic decrease in inference time of 90% enables the ETL models to perform inference in real-time, which is important for deploying on low-powered edge devices.

## 5 CONCLUSION

In this work, we have presented ETL: an efficient procedure for transfer learning of face tasks. ETL produces lightweight and accurate models for face tasks without large datasets by efficient pruning and transfer learning foundation face models. It has only one tunable hyperparameter, which adjusts the trade-off between compression ratio and accuracy, making it predictable and easy to use. The high compression ratio makes real-time inference on the CPU possible, which is essential for deploying deep models on low-resource edge devices.

## REFERENCES

- A. Dhall, R. Goecke, S. L. and Gedeon, T. (2012). Collecting large, richly annotated facial-expression databases from movies. *IEEE MultiMedia*.
- Antipov, G., Baccouche, M., Berrani, S.-A., and Dugelay, J.-L. (2017). Effective training of convolutional neural networks for face-based gender and age prediction. *Pattern Recognit.*
- Aytar, Y. and Zisserman, A. (2011). Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*.
- Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*.
- Bengio, Y., Bergeron, A., Boulanger-Lewandowski, N., Breuel, T., Chherawala, Y., Cisse, M., Erhan, D., Eustache, J., Glorot, X., Muller, X., et al. (2011). Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.
- Cao, Q., Shen, L., Xie, W., Parkhi, O. M., and Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE.
- Caruana, R. (1995). Learning many related tasks at the same time with backpropagation. In *Advances in neural information processing systems*.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*.
- Dua, I., John, T. A., Gupta, R., and Jawahar, C. (2020a). Dgaze: Driver gaze mapping on road. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Dua, I., Nambi, A. U., Jawahar, C. P., and Padmanabhan, V. N. (2019). Autorate: How attentive is the driver? *2019 14th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2019)*, pages 1–8.
- Dua, I., Nambi, A. U., Jawahar, C. V., and Padmanabhan, V. N. (2020b). Evaluation and visualization of driver inattention rating from facial features. *IEEE Transactions on Biometrics, Behavior, and Identity Science*.
- Duong, C. N., Luu, K., Quach, K. G., and Le, N. T. H. (2019a). Shrinkteanet: Million-scale lightweight face recognition via shrinking teacher-student networks. *ArXiv*.
- Duong, C. N., Quach, K. G., Le, N. T. H., Nguyen, N., and Luu, K. (2019b). Mobiface: A lightweight deep learning face recognition on mobile devices. *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*.
- Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv: Learning*.
- Gong, Y., Liu, L., Yang, M., and Bourdev, L. D. (2014). Compressing deep convolutional networks using vector quantization. *ArXiv*.
- Gourier, N., Hall, D., and Crowley, J. L. (2004). Estimating face orientation from robust detection of salient facial features. In *ICPR International Workshop on Visual Observation of Deictic Gestures*.
- Guo, D., Rush, A. M., and Kim, Y. (2021). Parameter-efficient transfer learning with diff pruning. In *ACL/IJCNLP*.
- Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer.
- He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. (2018). Filter pruning via geometric median for deep convolutional neural networks acceleration. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4335–4344.
- He, Y., Zhang, X., and Sun, J. (2017). Channel pruning for accelerating very deep neural networks. *2017 IEEE*

- International Conference on Computer Vision (ICCV)*, pages 1398–1406.
- Hitawala, S. (2018). Evaluating resnext model architecture for image classification. *ArXiv*.
- Houlsby, N., Giurghi, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *ICML*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*.
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. *ArXiv*.
- Iandola, F. N., Moskewicz, M. W., Ashraf, K., Han, S., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 1mb model size. *ArXiv*.
- Jin, H., Zhang, S., Zhu, X., Tang, Y., Lei, Z., and Li, S. (2019). Learning lightweight face detector with knowledge distillation. *2019 International Conference on Biometrics (ICB)*.
- John, T. A., Balasubramanian, V. N., and Jawahar, C. V. (2021). Canonical saliency maps: Decoding deep face models. *ArXiv*, abs/2105.01386.
- Khorrami, P., Paine, T. L., and Huang, T. S. (2015). Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*.
- Kim, M. and Smaragdis, P. (2016). Bitwise neural networks. *ArXiv*.
- Kossaifi, J., Tzimiropoulos, G., Todorovic, S., and Pantic, M. (2017). A few-va database for valence and arousal estimation in-the-wild. *Image and Vision Computing*.
- Lee, N., Ajanthan, T., and Torr, P. H. S. (2019). Snip: Single-shot network pruning based on connection sensitivity. *ArXiv*, abs/1810.02340.
- Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. (2017). Pruning filters for efficient convnets. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2016). Pruning filters for efficient convnets. *ArXiv*, abs/1608.08710.
- Lim, J. J., Salakhutdinov, R. R., and Torralba, A. (2011). Transfer learning by borrowing examples for multi-class object detection. In *Advances in neural information processing systems*.
- Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- Long, J. L., Zhang, N., and Darrell, T. (2014). Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*.
- Luo, J.-H., Zhang, H., Yu Zhou, H., Xie, C.-W., Wu, J., and Lin, W. (2018). Thinet: Pruning cnn filters for a thinner net. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2525–2538.
- Martin Koestinger, Paul Wohlhart, P. M. R. and Bischof, H. (2011). Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*.
- Miyashita, D., Lee, E. H., and Murmann, B. (2016). Convolutional neural networks using logarithmic data representation. *ArXiv*.
- Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. (2016). Pruning convolutional neural networks for resource efficient transfer learning. *ArXiv*, abs/1611.06440.
- Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*.
- Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In *British Machine Vision Conference*.
- Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*.
- Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*.
- Sharma, A. and Foroosh, H. (2020). Slim-cnn: A lightweight cnn for face attribute prediction. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*.
- Tommasi, T., Orabona, F., and Caputo, B. (2010). Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*.
- Wang, C. and Lan, X. (2017). Model distillation with knowledge transfer in face classification, alignment and verification. *ArXiv*.
- Wu, X., He, R., Sun, Z., and Tan, T. (2018). A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*.
- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*.

- Zamir, A. R., Sax, A., Shen, W., Guibas, L., Malik, J., and Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Zhang, H., Zhao, H., Liu, C., and Yu, D. (2020). Task-to-task transfer learning with parameter-efficient adapter. In *NLPCC*.
- Zhang, M. S. and Stadie, B. C. (2020). One-shot pruning of recurrent neural networks by jacobian spectrum evaluation. *ArXiv*, abs/1912.00120.
- Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. (2014). Object detectors emerge in deep scene cnns. *CoRR*.