# HEAD POSE ESTIMATION USING LEARNED DISCRETIZATION

*Se Yeon Kim*[⋆] *Scott Spurlock*[†] *Richard Souvenir*[‡]

[⋆] College of Computing, Georgia Tech
[†] Department of Computing Sciences, Elon University
[‡] Department of Computer and Information Sciences, Temple University

## ABSTRACT

We address the problem of automated discretization for continuous labels in the context of head pose estimation from overhead cameras. Due to the lack of visual detail, precise head pose estimates are not always possible. A common approach is to discretize the space of head pose angles, turning a real-valued prediction task into a coarser (ordered) classification variant. Often, however, the ranges are arbitrarily defined (e.g., dividing up parameter space evenly). Our work incorporates label discretization into the feature learning process and improves the accuracy of coarse head pan angle prediction from overhead cameras on a benchmark dataset.

***Index Terms***— pose estimation, supervised learning, clustering

## 1. INTRODUCTION

Automated gaze estimation has applications to surveillance and marketing. However, using typical overhead surveillance cameras, facial features (eyes, nose, mouth) may not be readily visible (Figure 1), limiting the opportunity for fine-grained gaze estimation. A common workaround is to discretize the pose space and return coarse predictions. For example, the cyclic head pan (left-right) range is commonly divided into eight evenly-sized bins, spanning 45 degrees of rotation each. However, such arbitrary partitions may not align with the natural change points inherent in the data. In this paper, we seek to learn coarse discretizations using a data-driven approach by incorporating label discretization into the training pipeline.

Approaches to head pose estimation can be broadly classified based on the apparent size of head in the image. For near-field applications, where faces occupy millions of pixels, recent methods (e.g., [1, 2, 3]) can provide fine-grained estimates and accurate facial marker predictions. However, we consider the far-field case, where a head may only occupy hundreds of pixels. Some methods rely on generic image feature descriptors (e.g., HOG, LBP) as part of a traditional machine learning classification pipeline [4, 5, 6], while others develop custom features [7, 8]. Similar to ours, recent approaches have incorporated convolutional neural networks (CNN) to sidestep the need for hand-crafted features [9, 10, 11]. However, these approaches rely on accurate face localization, which is not always possible with low-resolution face patches, as we demonstrate in our experiments.

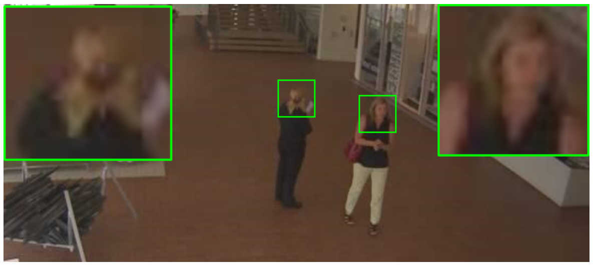

**Fig. 1**: Precise head pose estimates are not always possible from overhead cameras. Our method provides an alternative to arbitrary coarse discretization of head pose angles.

There have been some approaches designed for lower resolution images that do not discretize the label space, but provide real-valued estimates. One method operates on artificially-reduced resolution images for privacy preservation, but is designed only for forward-facing subjects [12]. Other approaches fuse estimates from networks of cameras to provide real-valued head pose predictions (e.g., [13, 14]). These approaches, however, depend on the availability of multiple, calibrated cameras in the environment.

While our work considers discretizing labels, previous work has considered discretizing feature values, or bucketization [15, 16]. For example, most variants of decision tree and Naive Bayes classifiers rely on categorical or ordinal features, so real-valued features are often converted to discrete analogs as a pre-processing step. Simple strategies, such as a uniform partitioning of feature space, tend to perform worse than methods such as Recursive Minimal Entropy Partitioning (RMEP) and error-based discretization. However, these methods rely on (accurate) labels as part of a supervised approach to discretizing features and, in the end, do not alter the prediction task. Label discretization, while superficially similar, is a much different task.

Our approach most closely follows recent strategies of in-

This material is based upon work supported by the National Science Foundation under Grant No. 1461166.

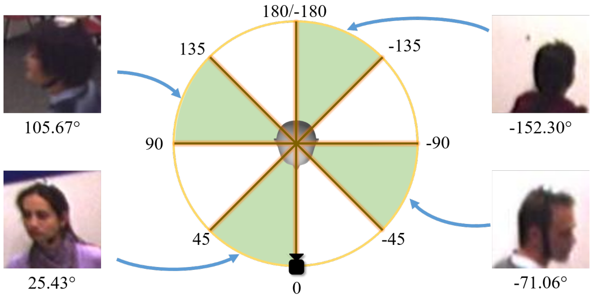

**Fig. 2**: For the discrete version of head pan angle estimation, each image, is assigned a label via a binning function.

corporating tasks beyond representation learning into an end-to-end learning framework, such as methods for simultaneous representation and image clustering [17, 18]. In the next section, we describe our joint approach for representation learning and label discretization.

## 2. METHOD

For head pose estimation from overhead cameras, training data is typically collected in calibrated, multi-camera environments with additional sensors (e.g., head trackers, depth cameras), which provide high-resolution head pose estimates, $\upsilon$. However, in the field, images typically lack the level of detail needed for fine-grained head pose estimates. For the discrete, coarse version (Figure 2), the goal is to predict $\mathbf{y} = \sigma(\upsilon; \Phi)$, where $\mathbf{y} \in \mathcal{Y}$ is a one-hot encoded vector of length $B$, corresponding to a bin in the discretized pose space, and $\sigma : \Upsilon \times \Re^B \rightarrow \mathcal{Y}$ is the binning function, with thresholds, $\Phi$. Previous methods using discretized labels typically employ binning functions that evenly divide the pose space or are otherwise arbitrarily-defined. In this paper, we incorporate learning the binning thresholds, $\Phi$, into the end-to-end learning process for discrete head pose estimation.

Similar to [14] and [13], we assume that head pose estimation is a step in an automated human activity analysis pipeline, and the head region of a tracked target has been roughly localized. So, given a training set of head image patches, $X$, and corresponding head pose parameters $\Upsilon$, for a query image, $x_i$, the goal is to predict the discretized head pose, $y'_i$ so that $y'_i \approx \sigma(\upsilon_i; \Phi)$. Initially, our method considers a fine-grained (i.e., large number of bins) classification task and alternates between: (1) representation learning for a fixed binning function and (2) decreasing the granularity of the binning function by merging the labels for ambiguous, adjacent bins.

### 2.1. Representation Learning

Let $f(\mathbf{x}; \Theta)$ represent a deep convolutional neural network (CNN) model with parameters, $\Theta$. Our approach is not specific to any particular model; for fine-tuning, it is common to start with a network trained for a related task. We adapt the network by setting the number of outputs to the number of discrete labels in pose estimation, with the bin thresholds given by $\Phi$. In the case of head pan angle prediction, most approaches (e.g., [5]) discretize the 360° angle space into eight evenly-sized, contiguous $45°$ bins. Training the network follows the typical supervised training approach for classification using labeled examples, minimizing the cross-entropy loss between the prediction and discretized parameter:

$$L(\Theta; \Phi) = \frac{1}{N} \sum_{i=1}^{N} H\left(y'_i, \sigma(\upsilon_i; \Phi)\right) \qquad (1)$$

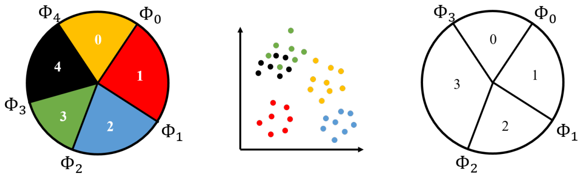

**Fig. 3**: The circles depict a binning function on a continuous, cyclic parameter space before (left) and after (right) merging. The feature representation shows the points colored by their current bin assignments. In this example, points from bins 3 and 4 have the highest affinity and will be merged, or, equivalently, bin threshold $\phi_3$ will be removed. (Best in color.)

### 2.2. Label Discretization

The second stage of our approach follows the merging step of agglomerative clustering, where a dataset of $N$ items is iteratively merged into a smaller number of clusters. Initially, each data item is its own cluster and the primary computation is calculating the affinity, $\mathcal{A}(C_a, C_b)$, between pairs of clusters, which characterizes the "closeness" of two point sets. At each iteration, clusters with maximum affinity are merged.

We adapt the merging step of agglomerative clustering to a supervised setting. Our feature representation for each example, $\mathbf{x}_i$, is the output of the network, $\mathbf{y}_i = f(\mathbf{x}_i, \Theta)$, a $B$-dimensional vector. Each example is assigned to one of $B$ clusters, corresponding to the ground-truth label and binning thresholds, $\mathbf{y} = \sigma(\upsilon; \Phi)$. Under this formulation, the affinity function, which measures cluster closeness, also serves as a measure of prediction ambiguity. That is, two clusters with high affinity correspond to examples from two different discrete labels (i.e., bins) which are assigned similar output representations by the network. High affinity clusters are good candidates to merge since the network cannot disambiguate the input examples.

Figure 3 demonstrates the merging process for a case where the original parameter is continuous and cyclic (e.g., head pan angle). This figure depicts both the binning function and the distribution of examples in feature space. Unlike the general case of clustering, for our problem, at each iteration, it is not necessary to consider all $O(B^2)$ pairwise affinities, but only those of adjacent bins, which can be represented by the $B$ bin thresholds incident to the adjacent bins. Many affinity measures have been proposed (e.g., [19, 20]). Empirically, we observed that different affinity functions yielded similar results. In our experiments, we used average-link clustering with the $L_2$ distance.

### 2.3. Algorithm

Essentially, our approach alternates between solving for classifier weights, $\Theta$, with fixed binning thresholds, $\Phi$, then updating the binning thresholds by merging. These steps iterate until the stopping criteria are met (e.g., desired number of bins). Algorithm 1 outlines our method.

**Algorithm 1:** Label Discretization

```
Input: labeled images, $\mathcal{X}$; associated labels, $\Upsilon$
Output: network weights, $\Theta$; bin thresholds, $\Phi$
1 Initialize $\Theta_0$ and $\Phi_0$
2 for $t$ in $1 \ldots T$ do
3     $\Theta_t \leftarrow \arg\min_{\Theta} L(\Theta, \Phi_{t-1})$     (Sec. 2.1)
4     Let $C_j \equiv \{\mathbf{y}'_i | \mathbf{y}_i[j] = 1\}$
5     $m \leftarrow \arg\max_j \mathcal{A}(C_j, C_{j+1})$     (Sec. 2.2)
6     $\Phi_t \leftarrow \Phi_{t-1} \setminus \phi_m$
7 $\Theta \leftarrow \Theta_T$
8 $\Phi \leftarrow \Phi_T$
```

The network weights, $\Theta_0$ can be initialized with weights from a similar problem or from scratch. For the initial bin thresholds, $\Phi_0$, we start with a relatively large number (i.e., 100s) of evenly-distributed bins as an initial fine-grained partitioning. For each iteration, $t$, the network is fine-tuned starting with the previous settings, $\Theta_{t-1}$. To reduce the number of times the network is retrained, we perform multiple merge steps (Lines 4-6 in Alg. 1) for each iteration of training the network. In Section 3, we evaluate the effect of these optimizations and the overall classification accuracy of our approach on the problem of head pose estimation from images.

## 3. RESULTS

We evaluate our method on DPOSE [4], a publicly-available dataset for head pose estimation, consisting of roughly 150,000 frames of 15 moving people captured by 4 calibrated cameras. These experiments focus on head pan angle estimation from roughly localized head image patches. The DPOSE data was split with images from 12 actors used for training and 3 used for testing.

The CNN was initialized following [21], a network designed for age estimation from facial images with three pairs of convolutional and max-pooling layers followed by three fully connected layers. We selected this CNN because it had been trained with many facial images [22]; however, our approach could be applied to other related networks (e.g., [23]) or trained from scratch with sufficient data.

|  | Ours | `Exemplar` | `Hyperface` | `HPE` |
|---|---|---|---|---|
| All | **68.09%** | 58.52% | 17.30% | 10.83% |
| Forward | **75.51%** | 61.45% | 32.88% | 20.64% |

**Table 1**: Classification accuracy for discrete head pan angle estimation on DPOSE data on both the full test and forward-facing subjects. Each bin represents a 45° range of angles.

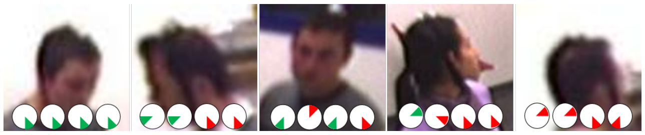

**Fig. 4**: Sample head pan prediction results. Glyphs show the pan angle prediction (0° pointed down, correct is green) for Ours, `Exemplar`, `Hyperface`, and `HPE`, respectively.

Our method was implemented in Python with Caffe [24] and trained on a standard PC with an NVIDIA K40 GPU. Input images were scaled to 227 × 227. Meta-parameters were selected by cross-validation on the training set. We used stochastic gradient descent (SGD) optimization with a learning rate of 0.0005 with a 10x multiplier on the last layer and a batch size of 250. Initially, training lasted 8 epochs. Post-merging rounds of training lasted 4 epochs.

### 3.1. Discrete Head Pose Estimation

To establish a baseline for the discrete classification, we compare our model to three recent methods for head pose estimation on an 8-way discrete classification task with evenly-divided bins: (1) `Exemplar` [14], which trains an ensemble of local exemplar SVM classifiers; (2) `Hyperface` [10], a multi-task CNN that incorporates fused features; and (3) `HPE` [6], which relies on probabilistic high-dimensional regression. `Exemplar` was trained with the same data as our baseline model, while `Hyperface` and `HPE` use pre-trained models. Table 1 shows the results for the 8-way classification task. Our method results in the highest accuracy for this task. Both `Hyperface` and `HPE` rely on face detection, so they fail in the case of rear-facing subjects. The second column of Table 1 shows the classification results using only test subjects corresponding to the 4 forward-facing bins, representing pan angles $[-90, 90]$. While the classification accuracy improves for all methods, the overall performance of `Hyperface` and `HPE` is still modest compared to our approach and `Exemplar`. We observed that face localization, which is an explicit step in these two methods, often failed in the case of low-resolution images, even for forward-facing subjects. Figure 4 shows example test images and predictions from each method.

| Merges | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Accuracy | 78.16% | 83.39% | 79.52% | 76.56% | 75.37% |
| Time | 32h | 16h | 8.5h | 4.6h | 2.7h |

**Table 2**: Classification performance and approximate algorithm run-time as the number of label merges per feature learning step increases.

| Method | Accuracy | Precision | Recall |
|---|---|---|---|
| Fixed labels | 68.09% | 69.10% | 67.35% |
| Learned labels | **79.52%** | **74.83%** | **74.64%** |

**Table 3**: Classification performance for our learned discretization approach.

### 3.2. Learned Discretization

Most discretization schemes for this task use 8 bins (most likely to correspond with the 8 cardinal and inter-cardinal directions), so we use $B = 8$ as our discretization parameter. Additionally, to reduce the amount of computation, we follow a typical pre-processing step in aggolomerative clustering with large data sets and perform an initial merging of nearby examples. In the case of discretizing labels, this corresponds to defining the initial bin thresholds, $\Phi_0$; we divide the $360^\circ$ degree label space into 90 evenly-spaced bins of $4^\circ$ each.

The primary meta-parameter to our approach is the number of steps of merging per round of re-training of the network, which reflects a trade-off between computational efficiency and accuracy of the resulting model. Table 2 shows the classification accuracy and approximate run-time of the method for an increasing number of merges per round of training. We selected 4 merge operations per training step, as it provided the best balance of accuracy and efficiency.

Table 3 shows the classification results using learned labels. Compared to the fixed approach, learned discretization results in bins of unequal size in parameter space and unequal numbers of training and testing examples in each class. To account for the imbalanced class issues, we report multi-class precision and recall (with macroaveraging). Overall, the gains with learned labels are consistent across all three metrics.

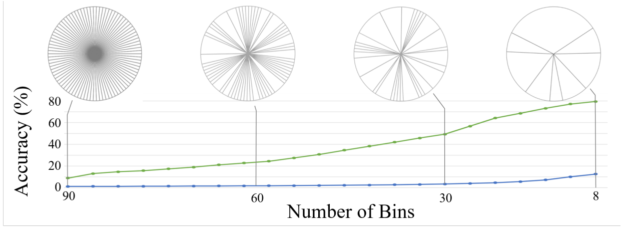

**Fig. 5**: (Top) Iterative label discretization from the initial (left) to final (right) states. (Bottom) Classification accuracy (green) during training compared to chance (blue).

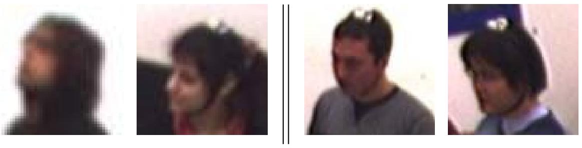

**Fig. 6**: The left images were assigned to a different bin than the right images by our learning process.

### 3.3. Discussion

Figure 5 shows snapshots of the iterative label merging process for one experiment and the evolution of test accuracy throughout the training process of merging bins. Due to the randomness inherent in many of the steps of our method, the learned bin thresholds (and image features) vary across different runs. However, for the task of head pan angle estimation, the distribution follows a similar pattern where rear-facing poses are more coarsely predicted than forward-facing poses. The angles represented at the learned bin edges tend to correspond to significant visual changes in the images. For example, Figure 6 shows example images on either side of a learned bin threshold. One potential explanation for learning this threshold could be that the change in visibility of the subject's right eye served as a discriminative visual feature. Examining visualizations of the network activations for borderline images may provide additional insight.

One observation is that, compared to the baseline CNN trained on the evenly-distributed fixed bins, our approach executes significantly more epochs of training over the course of the iterative merging and re-training stages. To examine this issue, we trained the initial network using the final learned bin labels with the same number of training epochs as our baseline CNN. The resulting network performed on par with the network learned after our iterative approach, suggesting that the bin thresholds, rather than the additional training iterations, leads to the increased prediction accuracy that is observed.

## 4. CONCLUSIONS AND FUTURE WORK

We demonstrated that classification performance could be improved using a novel data-driven approach to label discretization. Setting arbitrary discretization thresholds ignores the natural discontinuities in the image appearance. This approach is sufficiently general and could be applied to other learning-based vision problems where real-world phenomena are arbitrarily discretized, such as age or cloudiness estimation. One limitation to our approach is that (like other clustering methods) the number of target labels is pre-specified. We plan to investigate information-theoretic approaches to computing the optimal number of discrete labels as part of the learning process. Additionally, we plan to reformulate our multi-stage approach into an end-to-end recurrent framework.

## 5. REFERENCES

[1] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba, "Eye tracking for everyone," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2176–2184.

[2] Feng Lu, Yusuke Sugano, Takahiro Okabe, and Yoichi Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2033–2046, 2014.

[3] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling, "Appearance-based gaze estimation in the wild," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] Anoop K. Rajagopal, Ramanathan Subramanian, RaduL. Vieriu, Elisa Ricci, Oswald Lanz, Kalpathi Ramakrishnan, and Nicu Sebe, "An adaptation framework for head-pose classification in dynamic multi-view scenarios," in *Asian Conference on Computer Vision*, 2013, pp. 652–666.

[5] Yan Yan, Ramanathan Subramanian, Elisa Ricci, Oswald Lanz, and Nicu Sebe, "Evaluating multi-task learning for multi-view head-pose classification in interactive environments," in *Proc. International Conference on Pattern Recognition*. IEEE, 2014, pp. 4182–4187.

[6] Vincent Drouard, Silèye Ba, Georgios Evangelidis, Antoine Deleforge, and Radu Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *IEEE International Conference on Image Processing*, Sept. 2015, pp. 4624–4628.

[7] Javier Orozco, Shaogang Gong, and Tao Xiang, "Head pose classification in crowded scenes.," in *Proceedings of the British Machine Vision Conference*, 2009, vol. 1, p. 3.

[8] Diego Tosato, Michela Farenzena, Mauro Spera, Vittorio Murino, and Marco Cristani, "Multi-class classification on riemannian manifolds for video surveillance," in *Proc. European Conference on Computer Vision*, pp. 378–391. Springer, 2010.

[9] Ying Cai, Meng-long Yang, and Jun Li, "Multiclass classification based on a deep convolutional network for head pose estimation," *Frontiers of Information Technology & Electronic Engineering*, vol. 16, pp. 930–939, 2015.

[10] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *CoRR*, vol. abs/1603.01249, 2016.

[11] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, and Rama Chellappa, "An all-in-one convolutional neural network for face analysis," *CoRR*, vol. abs/1611.00851, 2016.

[12] Jiawei Chen, Jonathan Wu, Kristi Richter, Janusz Konrad, and Prakash Ishwar, "Estimating head pose orientation using extremely low resolution images," in *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2016, pp. 65–68.

[13] Rafael Munoz-Salinas, E. Yeguas-Bolivar, A. Saffiotti, and R. Medina-Carnicer, "Multi-camera head pose estimation," *Machine Vision and Applications*, vol. 23, no. 3, pp. 479–490, 2012.

[14] Scott Spurlock, Peter Malmgren, Hui Wu, and Richard Souvenir, "Multi-camera head pose estimation using an ensemble of exemplars," in *Proceedings of the 9th International Conference on Distributed Smart Cameras*. ACM, 2015, pp. 122–127.

[15] James Dougherty, Ron Kohavi, and Mehran Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. International Conference on Machine Learning*. 1995, pp. 194–202, Morgan Kaufmann.

[16] Ron Kohavi and Mehran Sahami, "Error-based and entropy-based discretization of continuous features," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996, pp. 114–119, AAAI Press.

[17] Zhangyang Wang, Shiyu Chang, Jiayu Zhou, and Thomas S. Huang, "Learning A task-specific deep architecture for clustering," *CoRR*, vol. abs/1509.00151, 2015.

[18] Jianwei Yang, Devi Parikh, and Dhruv Batra, "Joint unsupervised learning of deep representations and image clusters," *CoRR*, vol. abs/1604.03628, 2016.

[19] Anil K Jain, M Narasimha Murty, and Patrick J Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[20] Wei Zhang, Xiaogang Wang, Deli Zhao, and Xiaoou Tang, "Graph degree linkage: Agglomerative clustering on a directed graph," in *European Conference on Computer Vision*. Springer, 2012, pp. 428–441.

[21] Gil Levi and Tal Hassner, "Age and gender classification using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 34–42.

[22] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, Dec 2014.

[23] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[24] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM International Conference on Multimedia*, 2014, pp. 675–678.