Car Pose Estimation

Boyang Liu
Boston University

Douglas
Boston University

1. Problem Statement

Create a computer vision algorithm that can successfully predict the angle at which a picture of a car was taken at. For this given dataset and project, the angle 0 degrees is defined as the the front of the car being directly opposite the camera, 90 degrees is the front of the car pointing directly to the right, 180 degrees is the front of the car directly facing the camera and 270 degrees is the car pointing directly to the left. All angles in between represent states in between the above situations. Below in figure 1 are examples.

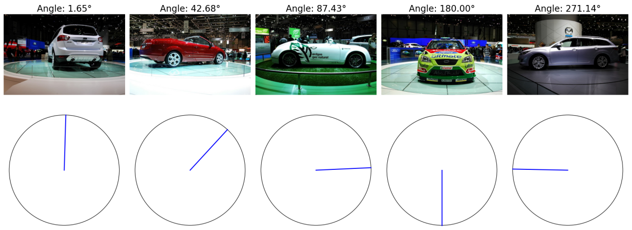

Figure 1. images and angles

2. Possible Applications

Possible applications for this algorithm include modeling cars in augmented reality/virtual reality, traffic surveillance and automated driving. In AR/VR, users could virtually 'walk around' and interact with 3D car models, enabling a dynamic viewing experience that mimics real life. For traffic surveillance, the ability to determine a car's orientation can enhance the monitoring and management of traffic flows, as well as improve the accuracy of automated incident reporting. In the realm of automated driving, accurately assessing the angle of surrounding vehicles provides essential information for navigation and collision avoidance, making it a critical component for the safety algorithms of autonomous vehicles

3. Dataset

The dataset used is a comprehensive collection of images showcasing 20 different cars, featuring a total of 2,299 images captured in sequences. Each sequence represents a full 360-degree rotation of a car, with images taken every 3-4 degrees. The data collection occurred at the Geneva International Motor Show '08, using a Nikon D70 camera mounted on a tripod with a Nikkor 12-24mm DX f/4 lens. Focal length remained constant for each car sequence but varied across different sequences, and the focus was manually set to the hyperfocal distance to ensure clarity throughout.

The dataset also includes text files specifying the bounding box coordinates for each image, and another text file labeled 'tripod-seq.txt' which offers valuable metadata, including sequence information, rotation angles, and rotation sense. This data is necessary for accurately computing the cars' rotation from the images. Lastly, the dataset includes timing files that correspond to the capture times for each image in a sequence.

The associated metadata specifies the exact number of frames captured to represent a complete 360-degree rotation. To determine the vehicle's orientation in any given frame, one would divide the 360-degree interval by the total number of frames that constitute a full revolution. This quotient yields the degree increment per frame. Multiplying this increment by the frame's ordinal position within its sequence, while adjusting for the direction of rotation (clockwise or counterclockwise), results in the precise angular displacement from the sequence's origin.

4. Angle Calculation

Since the images in the dataset do not come directly labeled, we had to label them ourselves. The determination of a car's angular orientation within the provided image dataset is facilitated by the detailed metadata that accompanies the sequences. Each car's initial position is not assumed to be uniform across the dataset; instead, the metadata files contain specific annotations for the frame number at which each vehicle is presented in a frontal pose. This pose serves as a zero-degree reference point for the origin of rotation. Additionally, the metadata details the total count of frames encompassing a complete 360-degree revolution for each vehicle, as well as the direction of rotation, with a positive value indicating a clockwise direction and a negative value signifying counterclockwise motion. Leveraging these data points, the angular position of the car in any image is calculated by identifying the frame's sequential number, subtracting the reference frame number to find the relative position, and then applying the degree increment per frame, which is obtained by dividing 360 by the number of frames per rotation. This calculation yields the car's rotation angle from the standardized frontal reference, ensuring a uniform approach to angle calculation across all sequences.

5. Method

In this section, we begin by defining the problem. Subsequently, we describe the baseline method and then detail our enhanced approach.

5.1. Problem Definition

In this study, we aim to predict the orientation angles of vehicles based on images of cars. Mathematically, the input consists of sequences of images $I \in \mathbb{R}^{n \times h \times w \times c}$, where each sequence corresponds to the predicted angles $A \in \mathbb{R}^n$ as the output.

5.2. Baseline

This task is framed as a regression problem. We employed a ResNet [2] architecture to extract features from the original images, followed by a simple linear layer to predict the angles directly. This approach is established as our baseline method.

5.3. Enhanced Approach

We observed that the task of predicting angles, which are extremely fine-grained, may present difficulties for the model in feature learning. To address this issue, we proposed a strategy to discretize the full 360° continuous angle range into nc classes. In this method, each continuous angle $a \in \mathbb{R}$ is mapped to an index $d \in \mathbb{R}$ in the discrete angle range set $S \in \mathbb{R}^{nc}$. This mapping transforms the problem into a more manageable classification task, thereby streamlining the learning process for the model. Following classification, we reconstruct the continuous angle based on the predictions of these discrete logits.

We implemented multiple discretization[1] to minimize the information loss associated with discretization processes. We also employed the mean shift method to reconstruct the continuous angle.

5.3.1 Multiple Discretization

We have divided the full 360° angle into nc classes, resulting in each angle range r being $\frac{360}{nc}$. Suppose we add a small shift t to the original angle a, we can then map $a+t$ into the discrete angle range set S. We introduce another hyperparameter ns, representing the number of sets of shifts. The unit shift t is defined as:

$$t = \frac{r}{ns} = \frac{360}{nc \cdot ns} \tag{1}$$

we can map the shifted angle $sa_i = a + i \cdot t$ to the index d_i for each discrete set, where i ranges from 0 to $ns - 1$. Consequently, the continuous angle a can be transformed into a set of indices $md = \{d_0, d_1, \ldots, d_{ns-1}\} \in \mathbb{R}^{ns}$, each corresponding to a different shift applied to a.

5.3.2 Mean Shift

Using the same ResNet model as the backbone for feature extraction, our method employs linear layers to transform the extracted features into logits $L \in \mathbb{R}^{ns \times nc}$. Each logit corresponds to the prediction of a discrete angle class associated with a shift angle. After computing the logits for various sets of shifted angles, we apply the softmax function to convert these logits into probabilities $P \in \mathbb{R}^{ns \times nc}$. These probabilities are then combined with the corresponding shifted base angles to yield the predicted continuous angle.

Given that it is inappropriate to directly multiply angles and probabilities, we initially transform the base angles into 2D positions using the transformations $x = \cos(\theta)$ and $y = \sin(\theta)$, where θ represents the base angles. The probabilities P are then multiplied with these 2D coordinates to compute the weighted average position across each set, given by $(\bar{x}, \bar{y}) = (\sum_i p_i x_i, \sum_i p_i y_i)$. Now $\bar{X}, \bar{Y} \in \mathbb{R}^{ns \times 1}$. Subsequently, the atan2 function is applied to convert this weighted position back into a set of angles.

Since these predictions are based on shifted angles, we adjust the results back to the initial base angles. Then, we can perform the $\alpha - (x, y) - \alpha$ operation once more to average the different set angles, thereby obtaining the final predicted continuous angle.

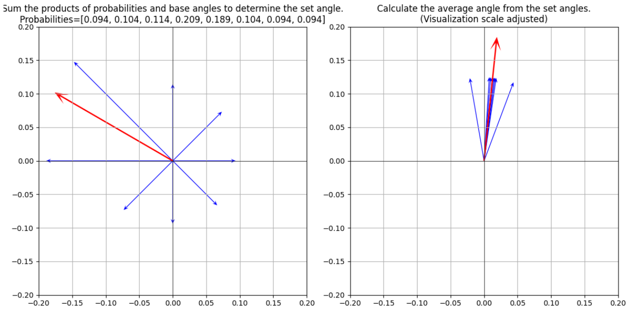

Figure 2. The left figure depicts the process of obtaining a set angle. The right figure illustrates the method for computing the average continuous angle from these set angles.

6. Experiments

In this section, we describe the dataset and detail our experimental procedures.

6.1. Datasets

We valid our methods using the EPFL GIMS08 dataset, which comprises images captured from varying angles of 20 distinct cars. We designate images of the first 10 cars as the training dataset and those of the remaining 10 as the testing dataset. The training set contains 1,179 images, while the test set includes 1,120 images. Each image is annotated with the car's angle, serving as the label. The dimensions of each car image are $[376, 250]$. Additionally, this dataset provides bounding boxes, which we use to crop the original images. Employing these cropped images significantly enhances both the performance and the convergence speed of our models.

6.2. Methods

For both baseline and our advanced methods, We employ a ResNet50 architecture as the backbone. We freeze the initial layers of the model and fine-tune only the last 15 layers. The learning rate is set at 0.001, and we utilize the Adam optimizer for training, which spans 30 epochs. To evaluate our models, we compute the Mean Absolute Error (MAE), chosen for its suitability and straightforwardness in evaluating the precision of angle estimations.

For the baseline method, which treats the problem as a regression task, we employ the Mean Squared Error(MSE) Loss for training. The final MAE for this model is 47.30.

In contrast, our advanced method conceptualizes the problem as a classification task. We utilize Cross-Entropy Loss for training the predictions of logits, achieving a final MAE of 35.40.

The training details are shown in figure 3 and figure 4

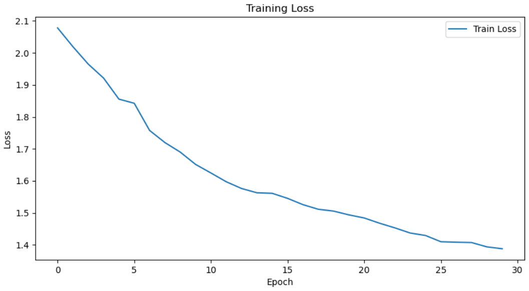

Figure 3. MSE loss in training

7. Results

Table 1 displays the final MAE performance.

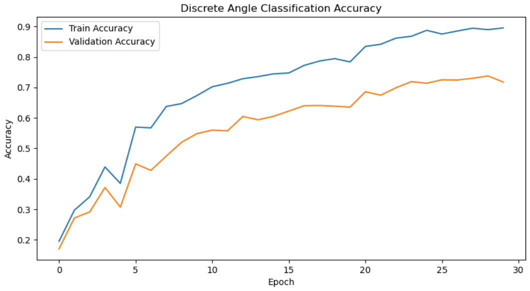

Figure 4. classification accuracy

Table 1. Comparison of MAE between Baseline and Advanced Method

Method	MAE
Baseline	47.30
Advanced	31.06

We also randomly selected some prediction results to display as shown in figure 5 below.

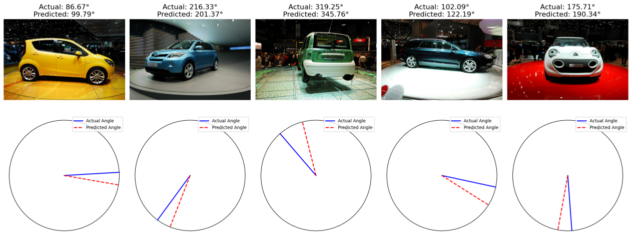

Figure 5. Input Images with ground truth labels vs predict

8. Conclusion

In this work, we focus on predicting car angles. We implemented multiple discretization methods to classify car poses into discrete angle ranges and used the mean shift technique to reconstruct the continuous angle predictions. We have proved the effectiveness of this method.

We are happy with the results we have obtained within the given time frame and constraints. Our model was able to obtain an MAE of 31.06. Future extensions of this project could experiment with different training datasets as well as different architectures. We believe that the dataset available was a limiting factor and with better, more diverse data, better results can be obtained.

Furthermore, our method does not rely on sequence information, enhancing its applicability across different scenarios.

Our backbone is relatively simple, allowing for future exploration with various computer vision models. Additionally, we observed that more complex mean shift methods, such as those [1] using the von Mises distribution as a kernel to calculate probabilities, could be investigated further.

References

[1] Kota Hara, Raviteja Vemulapalli, and Rama Chellappa. Designing deep convolutional neural networks for continuous object orientation estimation, 2017. 2, 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2