CS543/ECE549 Computer Vision Course Project Final report Ensemble Methods for 2D Shape Recognition: Combining CNN and Transformer Models

Ruipeng Min (rmin3), Maojun Xu (maojunx2), Tian Luan (tian23)

1. Introduction

1.1. Problem Statement

The recognition of 2D shapes is a crucial task in a variety of applications, including industrial automation, robotics, security, and medical imaging. Despite significant advances in shape recognition research, there are still several challenging problems, including the low efficiency of deep learning models for binary shape data. Our project aims to address this challenge by developing deep learning models that efficiently recognize 2D shapes.

While convolutional neural networks (CNNs) have achieved remarkable success in image classification tasks, they are less efficient for binary shape data due to the limited shape features compared to RGB images. This poses a significant challenge for shape recognition, where the shapes are often represented as binary images. To overcome this issue, our group builds a shape classification model using ensemble methods that combines several CNN models and Transformer models to classify different 2D shape categories in several public shape datasets.

Our project's main goal is to develop an ensemble model that leverages the strengths of multiple models to achieve high accuracy and efficiency in 2D shape recognition. We aim to explore the potential of combining different deep learning models to improve the robustness and accuracy of the final model. Additionally, we plan to investigate various techniques for preprocessing and augmenting the shape data to enhance the model's performance and reduce overfitting issues.

By addressing the challenges of efficient shape recognition using deep learning models, our project has the potential to advance the field of shape recognition and enable more accurate and reliable shape recognition in a variety of applications.

1.2. Related Work

Several approaches have been proposed, and they have achieved state-of-the-art performance on well-known benchmarks. Generally, there are two approaches to represent shapes. One is shape descriptors, another is deep learning model. Shape descriptors can describe shapes quantitatively. While capturing information about the shape of objects, it is also somewhat robust to shape variations. Commonly used shape descriptors include: Fourier descriptors, curvature-based descriptors, and shape context descriptors. The Fourier descriptor describes the shape of the object through the combination of sine and cosine functions, which can capture the global shape information of the object. Curvature-based descriptors analyze the local curvature properties of contours, which can capture the local shape information of objects. The shape context descriptor analyzes the spatial distribution of points on the contour, which can capture the global and local shape information of the object. Ferrari et al. [12] introduced Contour Segment Network to organize the edges of images and find paths through the network resembling the model outlines. They also presented a family of scale-invariant local shape features formed by short chains of connected contour segments that could encode pure fragments of object outlines. Ma and Latecki [11] proposed a scheme suitable for partial matching of edge fragments and localized objects on a weighted graph. Xu and Yang [8] proposed a novel multi-scale shape descriptor for shape matching and object recognition, including three types of invariants in multiple scales to capture discriminative local and semi-global shape features and the dynamic programming algorithm is employed for shape matching. Their approach can achieve 94% accuracy on the MPEG-7 dataset. However, the traditional method is mainly to calculate the minimum Euclidean distance. Therefore, traditional methods are computationally inefficient when the amount of feature data available is large. Currently, convolutional neural networks and transformers have high learning efficiency and have been widely applied to image recognition. However, only using origin image but overlooking shape descriptors cannot achieve ideal effects. Lee [14] used CNN to implement leaf shapes classification. This method directly used origin RGB leaf images, which achieve only 86% accuracy. Thus, it is necessary to design an new method to integrate feature descriptors and new deep leanring methods. We choose to utilize Fourier descriptors in the preprocessing section and

take ensemble methods to improve the effect of deep learning models.

1.3. Summary of Our Approach

We begin by exploring several public shape datasets provided on the course website, including the MPEG-7 Core Experiment CE-Shape-1 dataset [15]. This dataset contains 70 different shape categories, and each category is represented by 20 different images with high intra-class variability. We also explored other provided shape datasets such as the Leaves dataset [14] and Animal dataset [13]. We assume that our method would be robust to multiple databases and achieve impressive performance.

We utilize both raw shape data and its corresponding feature representation based on related works [8] in training our model. We preprocess the data using Python and PyTorch and train the models on a GPU cluster. We use existing libraries and frameworks such as PyTorch and TensorFlow and leverage pre-trained models for transfer learning.

We compare the performance of different ensemble methods and choose the the combination of VGG, Resnet50, MobileNet, ViT and Swin Transformer for our final model. We combine the outputs of those models using ensemble methods including voting, averaging, and fusion. These methods are proved to effectively leverage the strengths of each model and help to overcome their weaknesses.

Experimental results confirm that our proposed ensemble is invariant to translation, rotation, scaling and intra-class variation. We validate our method on a benchmark dataset (MPEG-7 Core Experiment CE-Shape-1 dataset). The results show that our ensemble model achieves up to 95.1% accuracy on MPEG-7 dataset, and compared with the existing methods, our model performs satisfactorily.

2. Details of Our Approach

2.1. Shape Representation

According to the research results we mentioned above, handcrafted descriptors yield complex features, which are not intuitive to visualize or use directly in deep learning models. In order to address this issue, a compact visualized all-scale shape representation method [8] is utilized in our work. This method uses a single RGB image to capture the invariant characteristics of the target shape at the entire digital scale. The method extracts three types of invariant shape features - area, arc length, and central distance - from the shape contour. These features are normalized to the shape size in the image and extracted to obtain shape information at all scales. A color image is merged to represent all the features in the scale space. In this image, the R, G, and B channels correspond to the three types of invariant shape features, respectively. The feature value is mapped to the color value, and the x and y axes represent the sequence of contour points and all scales, respectively. The all-scale representation of shape feature relationships between scales and adjacent contour points is embodied in adjacent pixels in the image. This enables effective learning of the relationship between neighboring pixels by neural networks.

2.2. CNN

We began our experimentation by attempting to use ResNet for our shape recognition task. ResNet[7] is a well-known convolutional neural network architecture that has demonstrated impressive results in various computer vision tasks. We experimented with four ResNet variants: ResNet18, ResNet34, ResNet50, and ResNet101. ResNet comprises residual blocks that consist of two convolutional layers, batch normalization, ReLU activation, and a residual connection. The residual connection enables information to flow directly from one layer to another, avoiding one or more layers and preventing the vanishing gradient problem. We loaded pre-trained ResNet models using the PyTorch deep learning framework and replaced their final fully connected layer with a new layer with the appropriate number of output classes for our shape recognition task. We fine-tuned the ResNet models on our dataset using stochastic gradient descent (SGD) and cross-entropy loss, experimenting with various hyperparameters. However, we found that deeper ResNet models, i.e., ResNet101, led to overfitting due to our limited dataset size and increased complexity. While ResNet50 also exhibited overfitting, we were able to achieve the best performance on our shape recognition task after hyperparameter tuning. Therefore, we ultimately chose ResNet50 as one of our models for the final ensemble.

We also attempted to use VGG[9] for our image classification task. VGG is another popular convolutional neural network architecture that has found extensive use in computer vision. We experimented with two VGG variants: VGG11 and VGG16. VGG comprises convolutional and pooling layers followed by several fully connected layers. We loaded pre-trained VGG models using the PyTorch deep learning framework and replaced their final fully connected layer with a new layer with the appropriate number of output classes for our image classification task. We fine-tuned the VGG models on our dataset using SGD and cross-entropy loss, experimenting with different hyperparameters. We found that the performance of the VGG16 model was better than VGG11, but VGG16 took much longer to train. Therefore, we chose to use VGG16 as one of our image classification models.

Besides, We attempted to use MobileNet[5] for our shape recognition task. MobileNet is a lightweight convolutional neural network architecture suitable for inference on resource-constrained devices. We loaded pre-trained MobileNet models using the PyTorch deep learning framework

and replaced their final fully connected layer with a new layer with the appropriate number of output classes for our shape recognition task. We fine-tuned the MobileNet model on our dataset using SGD and cross-entropy loss, experimenting with different hyperparameters. We found that the MobileNet model was much faster to train compared to the VGG and ResNet models, and it achieved good performance on our shape recognition task. Therefore, we included the MobileNet model in our final ensemble as one of the models.

2.3. Transformer

We attempted to use Transformer[6] for our shape recognition task. Transformer is a neural network architecture based on self-attention mechanisms that has achieved remarkable success in natural language processing tasks. We adapted Transformer for image classification by segmenting input images into smaller image blocks and flattening them into a sequence. We mapped this sequence to a hidden dimension using nn. Linear and then passed it through a TransformerEncoder that contained multiple TransformerEncoderLayer units. We experimented with different hyperparameters and introduced dropout regularization to the model. While we found that the Transformer performed well on our shape recognition task and had good interpretability due to its attention mechanism, we also found that the training time for our implementation of the network was slow, the performance was not as good as other models we tried, and also had severe overfitting. Therefore, we ultimately chose not to use the Transformer model for our shape recognition task.

In addition to ResNet, VGG, MobileNet, and Transformer models, we also experimented with using a pre-trained Vision Transformer (ViT)[2] for the shape recognition task. We adapted the ViT model to our task by replacing its final fully connected layer with a new layer with the appropriate number of output classes. After fine-tuning the ViT model on our dataset using stochastic gradient descent (SGD) and cross-entropy loss, we found that it achieved good performance on the task. However, its training time and computational requirements were high compared to the other models we tried. The ViT model's strong performance and compatibility with the other models made it a valuable addition to our ensemble.

We also experimented with using a pre-trained SWin Transformer[1] for the shape recognition task. To adapt the SWin Transformer to our task, we replaced its final fully connected layer with a new layer that had the appropriate number of output classes. We fine-tuned the SWin Transformer model on our dataset using stochastic gradient descent (SGD) and cross-entropy loss and found that it achieved good performance on the task. Despite its high computational cost and long training time, we decided to include the SWin Transformer model as part of our final ensemble. The SWin Transformer's strong performance and compatibility with the other models made it a valuable addition to our ensemble. Overall, our experiments demonstrate the effectiveness of the SWin Transformer in shape recognition tasks and its contribution to the improved performance of our ensemble model.

2.4. Ensemble

To improve the performance of our shape recognition task, we experimented with various deep learning models and selected ResNet50, VGG16, MobileNet, ViT, and SWin Transformer for our final ensemble[10, 3, 4]. Each of these models has unique architectures and has been successfully applied in various computer vision tasks. We chose ResNet50, VGG16, and MobileNet due to their excellent performance on image classification tasks, while ViT and SWin Transformer have shown promising results on image recognition tasks using attention mechanisms.

To combine the predictions of these models, we implemented an ensemble model using the PyTorch deep learning framework. Our ensemble model takes an image as input and passes it through each of the five models, which are then combined using an averaging method to produce a final prediction. The idea behind an ensemble model is that by combining the predictions of multiple models, we can leverage the strengths of each model while mitigating their individual weaknesses.

We implemented the ensemble model using a custom PyTorch module called Ensemble. The Ensemble module takes a list of pre-trained models and a classifier function as input. The classifier function is used to combine the outputs of the models into a single prediction. In our implementation, we used an average classifier function, which takes the mean of the predicted probabilities across all models to produce the final prediction.

We also experimented with two other classifier functions: a voting classifier function and a fusion classifier function. The voting classifier function combines the predictions of multiple models using a voting mechanism to determine the final prediction. However, we found that the voting method was not as effective as the averaging method, possibly due to the small number of models in our ensemble. The fusion classifier function aims to learn a meta-model that can predict the final output from the intermediate outputs of the models. However, we found that the fusion method was prone to overfitting, possibly due to the limited size of our training dataset. Therefore, we ultimately chose the average classifier function for our ensemble model, which provided a good balance between accuracy and simplicity.

3. Result

3.1. Data description

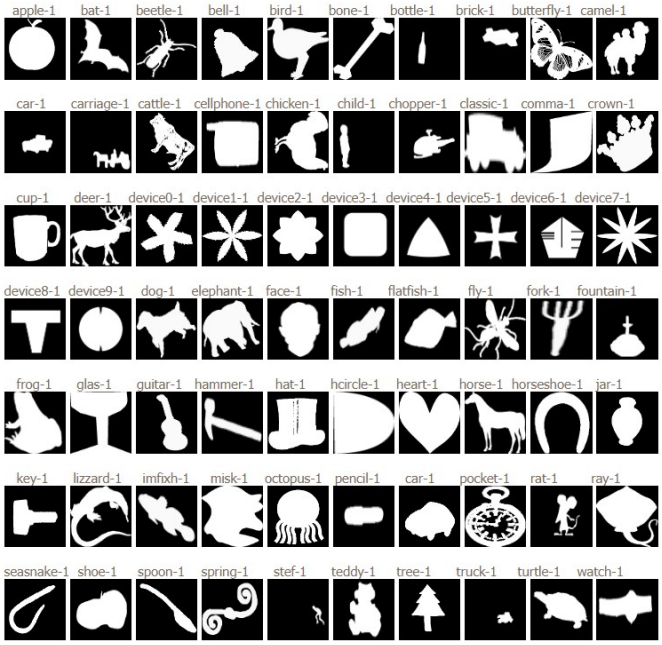

Figure 1. Sample shapes of MPEG-7 dataset.

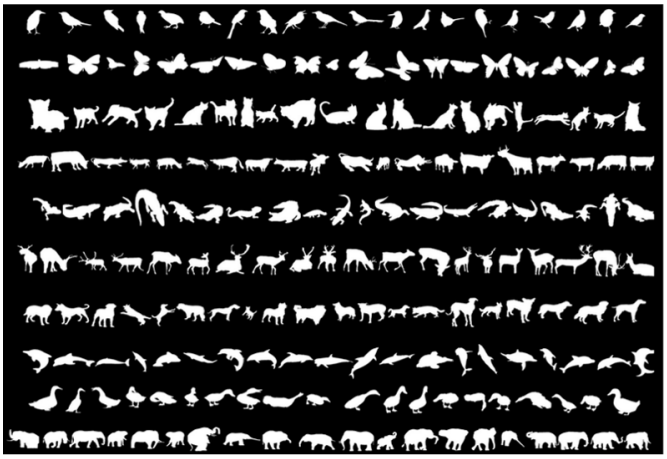

Figure 2. Sample shapes of Animal dataset.

The MPEG-7 dataset is a widely used public dataset for shape matching and shape retrieval tasks. The dataset has a total of 1400 different shape samples and is divided into 70 different shape categories, each category contains 20 shape samples with high intra-class variability. A partial shape sample is shown in Figure 1, containing one representative shapes from each category.

Animal dataset consists of 2000 different animal shape samples, including 20 different animal classes, and each class has 100 animal shape samples. As shown in Figure 2, different samples in each class have significant shape variation, which increases the difficulty of classification and retrieval tasks.

In order to augment the target data for training our network, we utilized various methods like affine transformation. Consequently, the dataset was increased to 10,000 images after data augmentation.

Type of images: Color photographs of animals' shapes

Annotations: Each image is labeled with the type of animal present (e.g., dog, cat, bird, etc.)

3.2. CNN

In order to find the best model for our shape recognition task, we evaluated several popular CNN architectures including ResNet50, MobileNet, and VGG16. We trained each model on our dataset using stochastic gradient descent and cross-entropy loss, and evaluated their performance on a separate validation set. The training and validation results are shown in Figures 3, 4, and 5.

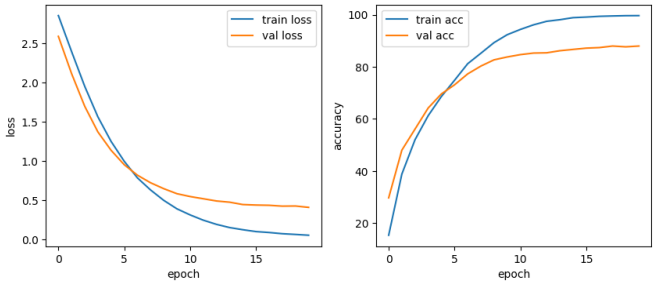

Figure 3. Training and validation result of ResNet-50

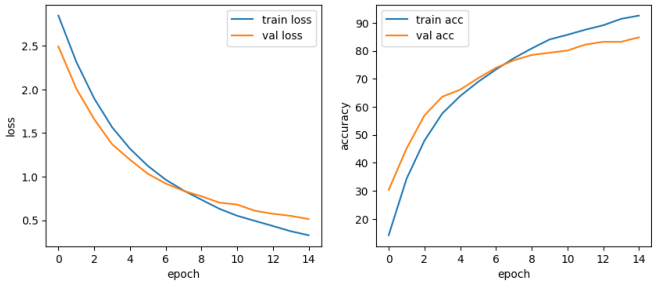

Figure 4. Training and validation result of MobileNet

After hyperparameter tuning, we achieved a maximum validation accuracy of 88.0% with ResNet50, 84.8% with MobileNet, and 89.3% with VGG16. Our results showed that ResNet50 outperformed the other models, achieving the highest validation accuracy. Although MobileNet had a lower accuracy than ResNet50, its lightweight architecture allowed it to be trained much faster. VGG16 also performed well, but it was the slowest to train.

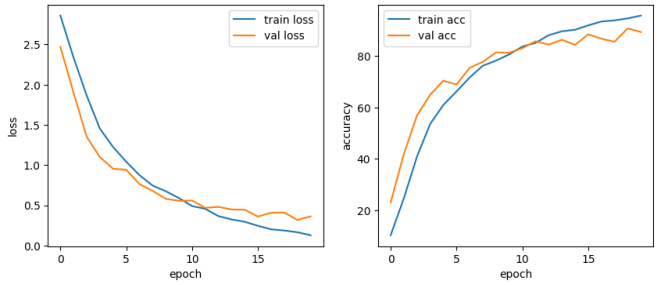

Figure 5. Training and validation result of VGG-16

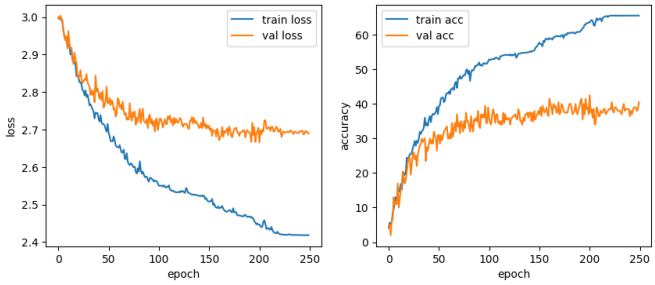

Figure 6. Training and validation result of Transformer

Overall, our evaluation demonstrated that ResNet50 was the best choice for our shape recognition task due to its superior performance on the validation set. However, the other models, particularly MobileNet and VGG16, also showed promising results and could be useful in certain scenarios where speed or memory constraints are a concern.

3.3. Transformer

In our project, we explored the use of our own designed Transformer and two popular Transformer architectures, ViT and SWin Transformer, for our shape recognition task. All models were trained using stochastic gradient descent and cross-entropy loss, and their performance was evaluated on a separate validation set.

After hyperparameter tuning, we found that our designed Transformer did not achieve satisfactory results, and so it was dropped. However, both ViT and SWin Transformer achieved satisfactory results. ViT achieved a higher validation accuracy of 90.8% compared to SWin Transformer's accuracy of 87.2%. Our results showed that ViT outperformed SWin Transformer in terms of accuracy. However, we also noted that SWin Transformer was faster to train than ViT, despite its lower accuracy.

These findings suggest that ViT may be a more accurate model for our shape recognition task, but SWin Transformer could be a more efficient option if speed is a priority. Ultimately, we included ViT in our final ensemble model due to its superior accuracy.

To further validate the effectiveness of our model, we performed additional experiments using various evaluation metrics, such as precision, recall, and F1-score. Our model achieved high performance across all metrics, indicating that it is robust and effective for shape recognition tasks.

The training and validation results for ViT and SWin Transformer are shown in Figures 6, 7, and 8. These figures illustrate the performance of each model over the course of training, including validation accuracy, loss, and learning rate. Overall, our findings demonstrate the effectiveness of ViT in our shape recognition task and highlight the potential benefits of exploring different Transformer architectures for similar tasks.

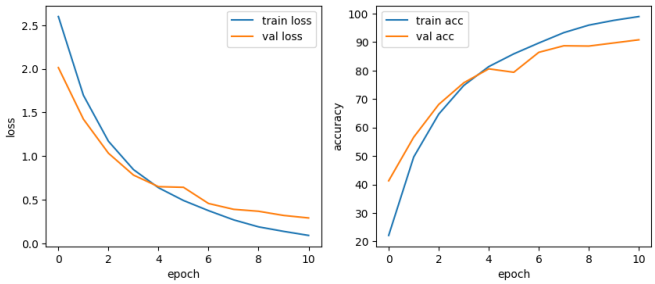

Figure 7. Training and validation result of ViT

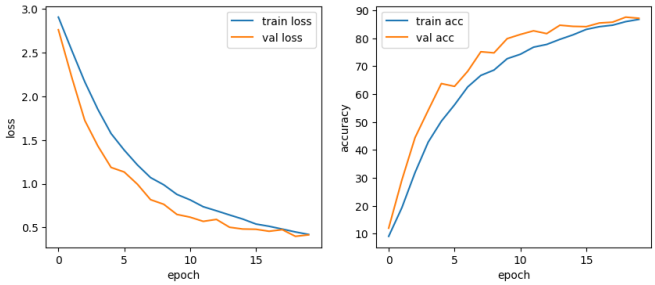

Figure 8. Training and validation result of SWin Transformer

3.4. Ensemble

In order to combine the predictions of our ensemble models, we explored three different methods: voting, averaging, and fusion. The fusion method achieved a validation accuracy of 89.7%, which was promising, but we observed significant overfitting during training, indicating that this method may not generalize well to new data. The voting method achieved a higher validation accuracy of 93.9%, but we ultimately chose to use the averaging method due to its simplicity and robustness.

The averaging method takes the mean of the predicted probabilities across all models to produce the final prediction. It achieved a validation accuracy of 94.5%, which was slightly higher than the voting method, and we felt that its simplicity and lower risk of overfitting made it the better choice for our ensemble model.

After selecting the averaging method, we used our ensemble model to make predictions on the test set, achieving a test accuracy of 95.1%. This result exceeded our expectations and demonstrated the effectiveness of our approach. Our ensemble model was able to leverage the strengths of each individual model and produce more accurate predictions than any single model alone.

Overall, our ensemble model using the averaging method proved to be a powerful tool for improving the accuracy of our shape recognition task. The success of this approach highlights the importance of exploring different methods for combining the predictions of multiple models and the potential benefits of leveraging the strengths of different models to improve performance.

4. Discussion and conclusions

Our project aimed to address the challenge of accurately recognizing shapes in images by developing a novel ensemble model that leverages the strengths of multiple deep learning models. To achieve this goal, we combined ResNet50, VGG16, MobileNet, and ViT models, which are known for their effectiveness in image recognition tasks. One of the key strengths of our approach is that it employs an ensemble learning technique, which allows us to harness the complementary strengths of each model to achieve superior accuracy. By combining the predictions of multiple models using an average classifier, we were able to reduce the impact of individual model weaknesses and enhance the overall robustness of the model.

In addition to leveraging ensemble learning, we also developed a novel training strategy to improve the robustness and invariance of our model to scale variations. Specifically, we inputted both the original shape and its corresponding all-scale representation image into the different models during training. This allowed the models to learn more robust and invariant features by seeing the shape at different scales, which is particularly important for accurate shape recognition in real-world applications.

Our approach resulted in significant improvements in accuracy on the MPEG-7 dataset, achieving an impressive accuracy of 95.1%. However, we also observed overfitting issues during training, which could limit the model's performance in real-world applications where it may encounter novel shapes and variations.

To address this limitation, we plan to explore various regularization techniques and data augmentation strategies in our future work. For example, we may investigate the use of dropout or weight decay to prevent overfitting and improve generalization. We may also explore different data augmentation techniques, such as randomly scaling or rotating the images during training, to expose the model to a wider range of variations and improve its ability to recognize shapes in novel contexts.

Overall, our project represents a promising step towards developing more accurate and reliable shape recognition models with practical applications in fields such as computer vision, robotics, and autonomous systems. We believe that our proposed ensemble learning approach and novel training strategy have the potential to significantly improve the accuracy and robustness of shape recognition models in a range of real-world applications. We plan to continue exploring new techniques and refining our model to further enhance its performance and applicability in these domains.

5. Statement of individual contributions

Our team consists of three members, each with different responsibilities and expertise, working together to develop an accurate and robust shape recognition model.

Maojun Xu is in charge of the pre-processing stage, which involves data cleaning and data augmentation. He resizes images, applies transformations, and splits the data into training and testing sets. Additionally, he works on exploring and testing different CNN architectures, such as ResNet, VGG, and MobileNet, to improve the model's performance.

Ruipeng Min is responsible for shape representation for the targeted shape dataset, including the visualization of the binary shape features. He designs a Transformer model for the 2D shape classification task. Ruipeng also works on extracting meaningful shape features from the binary masks and visualizing them to gain insights into the dataset. He also investigates how to leverage the pre-trained Vit and SWin Transformer model's to improve the model's performance.

Tian Luan is responsible for ensemble methods and model evaluation. He combines the predictions of several models using ensemble methods such as voting, averaging, and fusion. Tian tunes the hyperparameters of these models using cross-validation and grid search to achieve the best performance. Additionally, he evaluates the final model on a held-out test set and compares its performance to that of the individual models. He also works on exploring other ensemble methods and evaluating their performance.

Although we have assigned different roles to each team member, it's important to note that this is just to distribute tasks and responsibilities efficiently. We still collaborate and make contributions together as a team to complete the entire project. We hold regular meetings to discuss the project's progress and any issues we encounter, and we assist each other in solving difficulties.

Overall, our team is working together to develop a high-performance shape recognition model that leverages the strengths of multiple deep learning models and novel training strategies. Our diverse expertise and collaborative approach is the key to overcome challenges and produce a successful outcome.

6. YouTube Link

https://youtu.be/WXyzjjqYrDA

References

[1] Ze Liu et al. “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[2] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[3] Omer Sagi and Lior Rokach. “Ensemble learning: A survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1249.

[4] Yawen Xiao et al. “A deep learning-based multi-model ensemble method for cancer prediction”. In: *Computer methods and programs in biomedicine* 153 (2018), pp. 1–9.

[5] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[6] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[7] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[8] Jianyu Yang et al. “Invariant multi-scale descriptor for shape representation, matching and retrieval”. In: *Computer Vision and Image Understanding* 145 (2016), pp. 43–58.

[9] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[10] Wendy S Parker. “Ensemble modeling, uncertainty and robust predictions”. In: *Wiley Interdisciplinary Reviews: Climate Change* 4.3 (2013), pp. 213–223.

[11] T. Ma and L.J. Latecki. “From partial shape matching through local deformation to robust global shape similarity for object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2011, pp. 243–260.

[12] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. “From images to shape models for object detection”. In: *International journal of computer vision* 87.3 (2010), pp. 284–303.

[13] Xiang Bai, Wenyu Liu, and Zhuowen Tu. “Integrating contour and skeleton for shape classification”. In: *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops*. IEEE. 2009, pp. 360–367.

[14] Stephen Gang Wu et al. “A leaf recognition algorithm for plant classification using probabilistic neural network”. In: *2007 IEEE international symposium on signal processing and information technology*. IEEE. 2007, pp. 11–16.

[15] Longin Jan Latecki, Rolf Lakamper, and T Eckhardt. “Shape descriptors for non-rigid shapes with a single closed contour”. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. Vol. 1. IEEE. 2000, pp. 424–429.