Bare Demo of IEEEtran.cls for IEEE Computer Society Conferences

Junbo Liang
Student id: 1019905
Email: junbol@student.unimelb.edu.au

Yutao Zhou
Student id: 1001087
Email: yutazhou@student.unimelb.edu.au

***Abstract*—As image-based predictions become increasingly important in industrial automation and the stacking of objects in warehouses, how to improve the accuracy and efficiency of prediction models has become an urgent problem to be solved. Traditional models usually need to be trained in conjunction with real physical model modelling, which consumes a lot of manpower and material resources and is inefficient. In order to meet these challenges, this study proposes an improved model architecture that combines multiple Inception models to improve the accuracy of image-based predictions and the robustness of the model. The experiment was trained based on the ShapeStacks dataset and achieved a prediction accuracy of 62% on the test set. The research results show that the improved model architecture can quantify the performance of each model, can be quickly retrained to adapt to complex environments and put into use, has strong generalization ability, and has broad practical application prospects.**

1. Introduction

In modern computer vision, identifying object relationships between objects is very challenging. In this project, we will explore the importance of stability of stacked objects, design and implement image-based deep learning models to predict the stable height of stacked objects. We will use the ShapeStacks [1] dataset to train and validate the performance of our models. This dataset contains images of stacked objects of different shapes, colors, and arrangements. The challenge of this project is to understand the gravity and stability of objects through visual information. In this article, we will introduce different CNN pre-trained models and transformer models. Among them, the inception model and ViT model are the most outstanding. The inception network is a deep convolutional neural network architecture designed by the Google team. The main core idea is to use different sizes of convolution kernels (such as 1x1, 3x3, 5x5) to extract features in the same layer at the same time [2]. Unlike traditional convolutional neural networks (CNNs), the core of Vision Transformer is Self-Attention Mechanism. It divides the image into multiple patches of fixed size and then serializes these patches as the input of the Transformer. Om Uparkar and his team found that in medical image classification tasks (such as X-ray images), the pre-trained ViT model outperformed the CNN-based hybrid model on multiple evaluation metrics [3]. Here, we divided the entire experiment into two stages. In the first stage, we will explore which model works best in predicting object stability by comparing multiple models, such as transformer, ResNet, Inception and MobileNet. In the second stage, we will select the best model for subsequent experiments. Because the instability type of stack objects is divided into three categories, stable stack, unstable stack due to unsupported centre of mass and unstable stack due to stacking on non-planar surface, we decided to divide the data into different groups according to instability type. A total of 4 models were used to train the data. The first model was trained to predict instability type, and the other three models were trained to predict stable height of groups with different instability types, which greatly reduced the prediction difficulty of each model.

2. Experiment

2.1. Data Preprocessing

To ensure that the model can accurately predict the stable height of stacked objects, data preprocessing is a key step in the whole process. In the first stage, we first connect the images and data according to the id. In order to effectively evaluate the performance of the model, we further divide the original training dataset into training and validation sets. We use the commonly used train-validation split method, using 90% of the data for training and 10% for validation. In addition, we also use the stratification method to perform stratified sampling for different labels (stable height), which means that the category distribution of stable height will remain consistent in the training set and the evaluation set. This prevents some categories from being too few or missing in the training set or the test set. During the training process, we use the batch gradient descent algorithm to update the model weights with a certain number of samples each time. The choice of batch size is crucial to the training efficiency and performance of the model. After experiments, we choose 64 as the batch size because this setting achieves a good balance between computational efficiency and model stability. In the second stage, in order to further improve the training effect of the

model, we also introduced data augmentation technology, and performed Gaussian blur processing, edge detection, edge enhancement, translation, brightness change, horizontal flipping, and normalization processing on the images, expanding the diversity of training data, thereby improving the robustness of the model and preventing overfitting.

2.2. Baseline Method

In the first stage, in order to find the best performing models, we introduced four different models, namely Inspection model, ResNet model, MobileNet model and ViT model. And the last layer output of all models was replaced with 6 classes, in order to classify the categories into six categories corresponding to the stable height. In order to effectively train deep learning models, we designed a complete training process to ensure that the model can gradually converge and achieve good performance. This article will describe in detail each key step in the model training process, including the selection of optimizers, learning rate scheduling strategies, the use of data loaders, forward propagation and back propagation of models, and validation and model preservation strategies. First, we iterated the entire data 20 times to fully learn the characteristics of the data. We set the learning rate to 0.0005 to ensure that the update step size of the model parameters is moderate. We chose Adam optimizer because it can dynamically adjust the learning rate, which helps to speed up the convergence of the model. To ensure the repeatability of training, we set a fixed random seed to ensure that the same random numbers are generated each time. We input the image into the model, perform forward propagation, generate the output class prediction value, and then calculate the probability of each category through softmax, and select the category with the highest probability as the prediction result. The loss function we choose is cross entropy loss, which reflects the difference between the category probability distribution output by the model and the true category. The following is its equation. In order to deal with the problem of class imbalance, we assign different weights to each category according to the probability of the category appearing. The fewer the number of samples in the category, the greater the weight. During the training process, we will save the model weight with the highest validation accuracy to ensure the generalization ability of the model. We use validation accuracy and F-score as the evaluation criteria for model performance. Validation accuracy can well measure the overall classification performance of the model, while F-score combines precision and recall to better measure the performance of the model under the problem of class imbalance.

$$L(y, \hat{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

2.2.1. Result. Table 1 shows the Different Model Performances on Stability Prediction Task. From the table, we can see that the inception model has the highest F-score, reaching 0.625, which shows that it performs best in balancing precision and recall, while the f-score of the ViT model is only 0.1, which is much lower than other models, indicating that its prediction accuracy is the lowest. In terms of training accuracy, except for the ViT model, the accuracy of other models has reached more than 85%, indicating that they perform well on the training set and the models can learn the training data well. In terms of training loss, the training loss of ResNet and MobileNet is very low, 0.0002 and 0.0008 respectively, indicating that the models can fit the training data well. The training loss of the Inception model is slightly higher, at 0.0018, but still within an acceptable range. The training loss of the ViT model is 0.00252, which is significantly higher than other models, indicating that the model fits poorly on the training set. In the validation accuracy, Inception has the highest accuracy of 0.6276 on the validation set, indicating that it not only performs well on the training set, but also has strong generalization ability on the validation set, while ViT has the lowest validation accuracy of only 0.25, which verifies that its training and generalization abilities are both poor. As can be seen from Figure 1 and figure 2, the Inception model has the most balanced overall performance, with a high training accuracy, and the best validation accuracy and f-score among the four models.So we use the inception model as the baseline model.

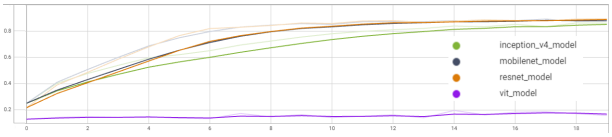

Figure 1. Baseline training accuracy

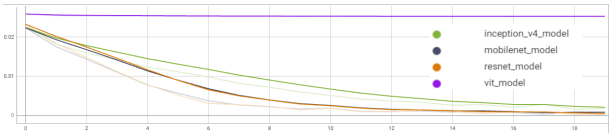

Figure 2. Baseline training loss

	ViT	ResNet	Inception	MobileNet
F-score	0.1	0.523	0.625	0.476
Training Accuracy	0.1599	0.8956	0.862	0.8798
Training Loss	0.0252	0.0002	0.0018	0.0008
Validation Accuracy	0.25	0.513	0.6276	0.478

TABLE 1. COMPARISON OF DIFFERENT MODEL PERFORMANCES ON STABILITY PREDICTION TASK

2.3. improved model

There are three types of instability in the dataset, namely stable, unstable due to misaligned centre of gravity, and unstable due to curved surface. Based on the idea of divide and conquer, we divide this task into two sub-problems: first predict the type of instability, and then predict the stable

height of the stacked objects accordingly, as shown in the Figure 3. Both the first and second step models use the Inception model, which we found to perform better on this task. The last layer of the first step model is changed to a linear layer with 3 classes representing the three types of instability. The input image will first predict the type of instability and then be passed to the corresponding second step model. The second step models are same as the best baseline model, but each of them have different weights so that they can better focus on the specific features to optimize the prediction accuracy, thereby improving the overall performance of the entire system.

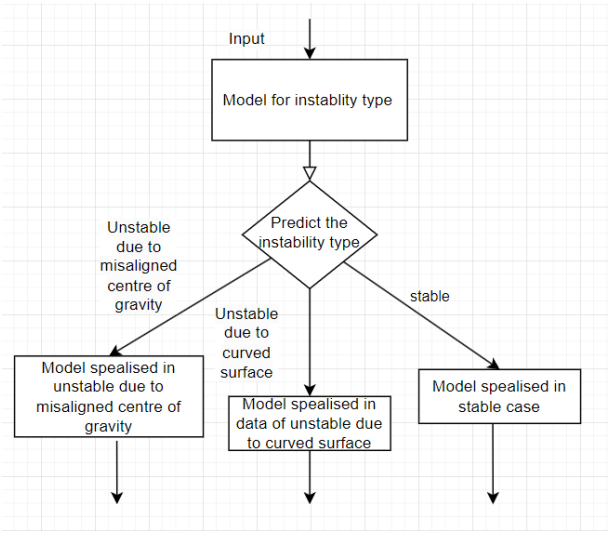

Figure 3. Flowchart

By training different models to focus on specific tasks, it is possible to quantitatively demonstrate the performance of each model and apply additional augmentations to improve specific models. In addition to this, if the system needs to handle a new type of instability in the future, it can be easily extended to accommodate this new situation by training an additional model for this new type of instability and retraining the first step model. This ensures that the performance is not affected by that of old instability types and speeds up the training process to adapt to new environments, thereby improving the generalization ability of the system.

model	Task	Batch size	Learning rate	validation accuracy
Model0	instability type	64	0.0005	74.5%
Model1	stable height for stable	32	0.0005	94.3%
Model2	stable height for central of gravity misaligned	32	0.0003	62.8%
Model3	stable height for curved surface unstable	32	0.0005	92.2%

TABLE 2. EXPERIMENT SETUP AND VALIDATION RESULT

As shown in Table 2, there are some parameters of each model set differently, and each model performs differently in the validation dataset. For Model 0, it is trained using 90% of the training dataset, thus using a relatively large batch size to ensure that the class distribution is similar to the original dataset. For Models 1 to 3, they are trained using data classified into their corresponding instability types, which are smaller datasets than the one used in Model 0, so they use relatively small batch sizes to prevent the risk of overfitting. We performed data enhancement on the trained dataset to allow the model to learn more from the same image, thus preventing the model from overfitting easily. Combined with a lower learning rate, each model was trained for 40 epochs to ensure that they converged, and in the process found the model weights that achieved the best performance on the validation set. For the validation dataset, no data enhancement was applied in order to maintain its properties to better represent the test data.

2.3.1. Result. All models were trained for a full 40 epochs, and because the model weights were saved when the highest validation accuracy was achieved. Thus, even if the model ended up overfitting the training dataset, this would not affect the model weights we used for prediction. From Figure 4 and Figure 5 we can see that all models have an accuracy of more than 90% on the training dataset, while the loss is less than 0.005, which means that our models are converged in the right direction because they all extract those important features from the images and correctly learn the relationship between these features and the results. After 25 epochs, Model 0, Model 1, and Model 3 all converged as their loss curves flattened out; after 35 epochs, Model 2 also converged.

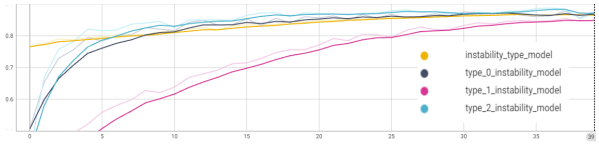

Figure 4. Train accuracy

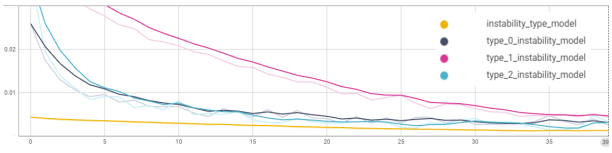

Figure 5. Train loss

While theory suggests that the performance of the entire system can be improved by splitting the task into two parts and training the models to focus on their specific task, the results deviated from our expectations. Model 2 only achieved 62.8% accuracy on the validation dataset, which should have similar characteristics to image in the test dataset. According to the Bucket Theory, the performance of our system depends on the poorest performance part, which is the Model 2, so the overall performance will not

be high enough. Although the validation accuracy of Model 1 reached 94.3% and that of Model 3 reached 92.2%, which are close to 100%, due to the poor performance in Model 2, the overall accuracy on the test dataset is only 60.2%.

Model 2 aims to predict stable heights for patches with misaligned centers of mass, which is also a challenging task for humans because some key information is hidden from the input image due to viewpoint. Even if the Inception model successfully focuses on the correct region in the image, it still cannot compute whether these blocks are stable or not. Compared to Model 1, which specializes in the stable case, and Model 3, which specializes in the unstable case of surfaces, Model 2 has a much more difficult task. Even though we tried more data augmentation and different learning rates, the best validation accuracy is still not high enough. Additionally, the validation accuracy of Model 0 is not as high as we expected, at only 74.5%, and most of the discrepancy is due to misclassified the stable cases and the unstable cases due to misaligned centre of mass. Therefore, our overall model is not as significantly improved as we expected due to the probability product.

3. Conclusion

We investigate the stable height prediction of stacked blocks in the context of vision-based model. We experimented with a single model to perform this task and found that the Inception-v4 model achieved the highest performance, so we set it as the baseline model. We then developed a new method to split the whole task into two tasks, first predicting the unstable state of the image, and then predicting the stable height of the stacked blocks using a specific model corresponding to the specific unstable state. This report describes the experimental setup as well as the results of our improved model. Although the prediction accuracy of the improved method on the test dataset is not significantly improved compared with the baseline model, when there are more datasets and more instability types in the new application environment, the improved method can ensure shorter retraining and fine-tuning time, so the improved method has higher generalization ability.

Our improved method does not perform well in predicting the stability of stacked blocks and predicting the stable height of stacked blocks with misaligned centre of gravity, which are the two main reasons for the relatively low performance on the testing dataset. We can develop another model to generate images from different angles of the input image, so that our improved model can learn the same features but from different viewpoints, achieving a more accurate analysis of the features through perspective voting. We can use more pre-processing methods to eliminate the influence of the background in the input image by removing texture and edge information.

By improving the model architecture, this work not only addresses a possible solution to current challenges, but also paves the way for future development of a robust and scalable system. Its adaptability in complex environments can be widely used in automated systems for handling stacking of items in various industries.

References

[1] Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. Shapestacks: Learning vision-based physical intuition for generalised object stacking, 2018.

[2] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[3] Om Uparkar, Jyoti Bharti, R.K. Pateriya, Rajeev Kumar Gupta, and Ashutosh Sharma. Vision transformer outperforms deep convolutional neural network-based model in classifying x-ray images. *Procedia Computer Science*, 218:2338–2349, 2023. International Conference on Machine Learning and Data Engineering.