Replies to Critical Reviews

Critical review from **group 3**

- Did you learn anything from your baseline models' most important features? Or did you use this 62.25% to compare against anything? It was mentioned early in the presentation, but it was never brought up again later.
  In fact, the baseline model is used to evaluate whether our deep learning methods have improved the prediction accuracy or not. Just like a basic standard. In our paper, we will make a comparison over the result obtained from the baseline model and deep learning model.

- Was PCA only used for the Transformer model? Did you try it for other models?
  In our project, PCA is a way to preprocess the images and it can be used for any three deep learning models we used. But in the ResNet model we have used some feature extraction ways to do the preprocess. And Vit models have used this and have got a better result. But when it comes to the InceptionV3 model, the PCA makes the result worse. In my view, it is likely that the PCA preprocessed will make the images lose some features and when it applies to the inception V3, the result is not desirable.

- There is slight confusion about what pre-processing steps correspond to which model, so recommendation is to be very clear about the feature extraction only being for ResNet and PCA for transformers. Slides did not mention any pre-processing for InceptionV3.
  When it comes to the InceptionV3 model, the PCA makes the result worse. In my view, it is likely that the PCA preprocessed will make the images lose some features and when it applies to the inception V3, the result is not desirable. In our inceptionV3 model, we have made some basic pre-processing, such as rotation, zoom and shift.

- One suggestion is that you should standardize the way you are displaying your loss and accuracy curves across your ResNet and Inception experiments. Since they are the same task, comparing the curves could be an interesting point of discussion. Inception has no inclusion of loss/accuracy curves in your presentation.
  Thank you for your suggestion. Due to the limitation of GPU resources, it is hard for us to display our loss and accuracy curves across all the models, but we will use tables to display our results and put them together to make a comparison.

- Another suggestion is to include examples of correct and incorrect predictions of all of your models so we can see potential sources of error with the models other than InceptionV3.
  Thank you for your suggestion. Due to page limitation, we cannot include this part for discussion. But with our observation on the test result, we found that types with apples tend to have low accuracy. Possible reason could be that in the dataset, apples are classified into 12 different types. Between some types the features are very similar.

- Did you test training the ViT for more than one epoch? What were the results like?
  Yes, for the ViT model, I tried it with three epochs. The performance increases significantly, especially for the average training accuracy. The average training accuracy is 99.7% and the average validation accuracy is 98.1% now. More detailed data are in the project report.
- For InceptionV3, the validation accuracy is 8% higher than the training accuracy, so do you have any explanation for this? Usually training accuracy is higher or close to validation accuracy, so this is an interesting discovery.
  Yes, it is an interesting question. In our presentation, I have trained about 10 epochs to get the result and the training accuracy is close to the validation accuracy. I think this phenomenon is caused by uneven distribution of the data and less amount of training epoch at first. But as the training epoch increases, the result seems to be reasonable.

- There was no conclusion about which model you found was the best. It seems ResNet had the highest accuracy of all 3, so maybe you can investigate and hypothesize why it outperformed the others.
  Thanks for the suggestion, since more tuning and adjustment is needed for the three models to reach optimal performance, we will put the final performance in the conclusion part of the report.

- In your code portion, you mention sampling to create a smaller dataset, but did you do anything to maintain class balance?
  Yes, when creating a smaller dataset for ResNet50 feature extraction, a function called get_sample is used to ensure every class of images is extracted with the same portion.

Critical review from **group 8**

- What is the intuition behind choosing these models?
  Random forest, ResNet50, Vit, inceptionv3 are very popular models in the prediction tasks. And all of them have very high accuracy in the ImageNet model. Vit models already have several pretrain models and only few tuning is needed when applied to our dataset.

- What are the main differences in the results between these models? Aside from the accuracy numbers, do they have noticeable differences in behavior in predicting new images?
  The main differences are reported in the *Project Analysis* in *Discussion* section. Besides the testing accuracy, we also compared the loss and F1 score of the models. The models all have good accuracy in predicting images. However, the complexity of them is not the same thus leading to different suitability to real-time classification. In our cases, the VIT model is the most suitable one.

- The intuition of using ResNet50 for feature extraction. Why do we need this step?
  Using a pre-trained model to do classification is widely used and often with high accuracy. Besides, applying the features extracted by ResNet50 to fully-connected-layers can reduce the trainable parameters, which increase the training speed.

- Is there a possible explanation of why the accuracy of validation sets trained with both models are so high?
  Because the parameters of the model have been pre-trained. And we can get a very high validation accuracy on our models.

Critical review from **group 36**

- Is there a reason to use Random Forest as the baseline? Is it because of its flexibility?
  Random forest is a very common way for prediction tasks and it is a machine learning method and pretty simple. So we can make a comparison of our deep learning models over the simple one. In this way, we can evaluate whether our model can improve the prediction accuracy .

- Maybe add the underlying principles of PCA and visualize the principal component for some images?
  Yes, PCA is used for dimensional reduction and optimal reconstruction. In this project, we first generate a set of images from one sample image with random mean Gaussian, random flipping and random rotation. Then, apply PCA on three channels separately and reconstruct the image from 8 principle components. The resulting image is used for further training.

- Explain more details about the InceptionV3 model?
  Yes, we will do this in our paper.
  Inception V3 is a very popular mode in the prediction task. The main idea of the Inception architecture is to find out how to use dense components to approximate the optimal local sparse nodes. There are several versions of the inception model, including inception, inceptionv2 and inceptionv3. InceptionV3 is the improved version of inceptionV2. Inception V3 has made two main improvements to Inception V2. First of all, Inception V3 optimizes the structure of the Inception Module. Now Inception Module has more types (there are three different structures: 35 × 35, 17 × 17 and 8 × 8), and Inception V3 is still in the Inception Module Branch is used in branch.Secondly, Inception V3 also introduced the method of splitting a larger two-dimensional convolution into two smaller one-dimensional convolutions. For example, 7×7 convolution can be split into 1×7 convolution and 7×l convolution. 3x3 convolution can also be split into lx3 convolution and 3 × l convolution. This is called Factorization into small convolutions though. In the paper, the author pointed out that this asymmetric convolutional structure splitting can handle more and richer spatial features and increase feature diversity. It can be better than symmetrical convolutional structure splitting, while reducing calculation amount.There are totally 295,683 trainable parameters in the model.

GROUP 12: FUIT RECOGNITION

Chong He, Mingen Li, Entong Su

University of California San Diego, La Jolla, CA 92093-0238,

ABSTRACT

In this project, we intend to better understand machine learning and deep learning through making a classification over the fruits and vegetable. Feature Extraction with Resnet50, Vision Transformer (ViT)and InceptionV3 are applied to Fruits-360 dataset to train the classifier. The performance of models are compared with each other and ResNet50 achieves the best result.

***Index Terms*—** fruit recognition, fruit detection, machine learning, deep learning

1 Introduction

As the development of science, the application of deep learning or machine learning methods on the image processing, such as Semantic segmentation, image classification becomes more and more popular. Our project focuses on how to use deep learning algorithm to perform fruit classification. Fruit classification can be used in a wide range of area like unnamed market and agricultural application in inter-cropping. This system is very necessary due to the large requirement of fruits and vegetable and high cost of manual labor.

In our project, the input is the images of 131 different kinds of fruits and vegetables associated with their labels from Fruits-360 dataset. We use PCA to preprocess the images for ViT. Machine learning methods, such as random forest and deep learning methods, such as ResNet50, ViT, InceptionV3 are applied for training, to predict the types of vegetable and fruits.

2 Related work

To perform fruit recognition, researchers describe fruit attributes with quantitative measurement and formulate the controlled vocabulary and equation to include traits features [1]. They use the Tomato analyzer to analyze specified shape features and combine them with PCA and subsequent QTL analyses. Thus, accurate measurement is provided for both fruit shapes and traits. However, massive effort are required for manually labeling and quantifying different traits and shape attribute for different plants.

Some researchers perform random decision forest algorithm on sorting papaya into different ripeness stage [2]. Decision trees processes and outputs the most probable class based on its model and learns the influence of input features. This algorithm also performs well in foods quality evaluation [3]. However, random decision forest can have a complicated structure and thus requiring massive computation power.

With the consideration of low efficiency and high time consumption of classifying the fruit by hand, CNN-based models become more and more popular used in the field of object classification. For example, in [4], Zhai *et al.* scale ViT models and train a classifier, which obtains a new state-of-the-art on ImageNet of 90.45%. The work in [5] aims at classifying vegetable and fruits in the retail market, sharing similar goal with our project. The authors use different conventional network such as Inception and MobileNet. Based on the works mentioned above, it is clear that the proposition of different deep learning based method has greatly simplified the object classification problem with abundant data. Traditional methods need lots of statistical and mathematical analysis. In contrast, deep learning based method can help us formulate the model with simple math. In spite of the advantage mentioned above, there are limitations too. For example, deep learning is unable to encode the position and orientation of object and lack of ability to be spatially invariant to the input data.

3 Dataset and Features

3.1 Dataset

Fruits-360 is first proposed in [6] and is continuously maintained by the authors. Till now, it has 90380 images of 131 fruits and vegetables with training set size of 67692 and testing set size of 22688. This dataset can be downloaded from the addresses pointed by references [7] and [8]. Every image in this dataset is with 100×100 pixels. Different from many other image dataset, the backgrounds of the pictures are removed in Fruits-360 as shown in Figure 1. In fact, this dataset can be used to conducted different image tasks, such as semantic image segmentation, object detection and object classification. The reason why we choose classification as our task is that this dataset has high-resolution images with little noise, ensuring a highly efficient classifier. For the excessive

data and limited personal computational resource, only part of the data in the training set is used for the training, validation and testing steps. 20000 images in the training set with the same portion of images in each class are used. The training, validation and testing sets are split by the *train_test_split* function in Sciki-Learn toolkit [9].

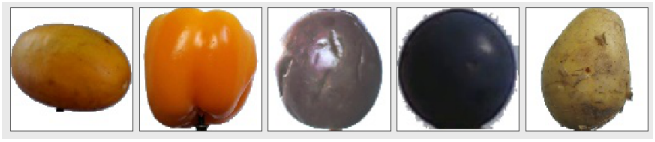

Fig. 1: A visualization of 5 random samples in the training set of Fruits-360.

3.2 Features

ResNet50: When it comes to the feature extraction with Resnet50 [10], the model pretrained on the ImageNet [11] is used. The 2048 dimensional feature vector are obtained by removing the last fully connected layer. With these high-efficient features, simple deep learning structure can achieve good classification accuracy.
PCA: Principle Component Analysis is a method for optimal reconstruction and dimension reduction. Data can be project into new orthogonal coordinate system with coordinate being the principle component, the feature reconstructed on the basis of the original data feature. The component with greatest variance is assigned to be the first principle component. With various noise and transformation applied, PCA is applied to generated set and outputs an image with 8 principle components. The processed image is used for further training.

Fig. 2: Image before (upper) and after (lower) processing with PCA

4 Methods

4.1 Random forest

Random forest is composed of many independent decision trees.When we classify, the new input sample will let the decision tree in the forest make a judgment and each decision tree will get its own classification. Model outputs the class with most selection by the decision trees, then the random forest will regard this result as the final result. But the random forest is unavoidable to overfit in some noisy classification or regression problems.

4.2 ResNet50

To ease the difficulty of training a very deep network, He *et al.* presented a residual learning framework in [10]. The layers of ResNets are reformulated as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. This fundamental breakthrough allows later generations to train extremely deep neural networks with 150+layers successfully. The plain baselines of the residual net are mainly inspired by the VGG nets [12]. Based on the plain network, after replacing each 2-layer block with a 3-layer bottleneck block, 50-layer ResNet is formulated, which is the structure implemented in this project. The structure is shown in Fig. 3. Instead of training the network from the scratch. ResNet50 pretrained by ImageNet is used to extract the feature. Those features are then applied to one fully-connected layer with softmax to classify the images.

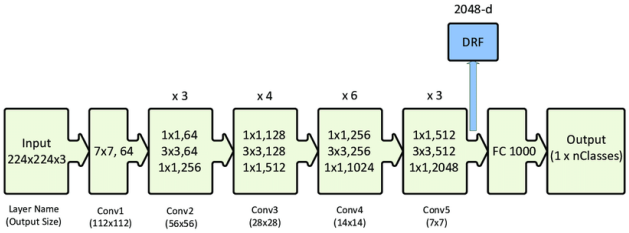

Fig. 3: The structure of ResNet50 Network [13].

4.3 Vision Transformer

Vision transformers (VIT) [14] utilize the transformer model structure with several data processing. An 2D image $x \in \mathbb{R}^{H\times W\times C}$ with dimension height H, width W and number of channel C is resized into a sequence of 2D patches x_p with dimension $\frac{HW}{P^2} \times P^2C$ where P define the resolution of the patches and the transformer will has N patches as input, each patch will be transformed into a 1D vector with length D as equation 1.

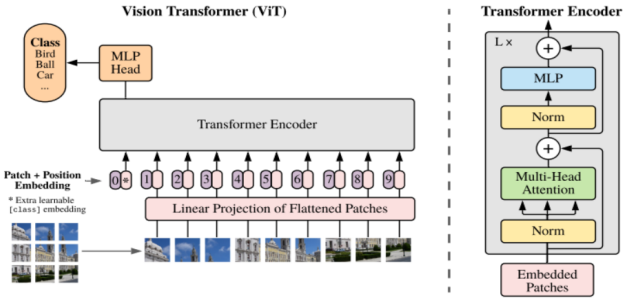

Fig. 4: Structure of ViT and transformer encoder [14]

Additionally, Since positional information for patches is still not presented to the model, a positional embedding is appended in figure 1 in purple block. A learnable embedding for the class patches x_{class} which will be used for the output of transformer encoder in equation 2. $z_L^{'}$ will be used as the image representation after going through the encoder. A classification head which is implemented by Multi Layer Perceptron (MLP) is applied to this output.
The encoder structure is similar with the transformer encoder used in NLP [15]. For layer l, with input z_{l-1} from previous layer, it first go through a multi-head self-attention layer in equation 3. Then a Multi layer Perceptron layer is applied in equation 4 and output the z_l. Layer normalization is performed before entering MLP and MSA.

$$z_0 = [\boldsymbol{x}_{class}; \boldsymbol{x}_p^1 E; \boldsymbol{x}_p^2 E ...; \boldsymbol{x}_p^N E] + E_{pos} \quad where \quad E \in \mathbb{R}^{P^2 C \times D} E_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

$$\boldsymbol{y} = LN(\boldsymbol{z}_L^0) \tag{2}$$

$$\boldsymbol{z}_l^{'} = MSA(LN(\boldsymbol{z}_{l-1})) + \boldsymbol{z}_{l-1} \quad l = 1, ..., L \tag{3}$$

$$\boldsymbol{z}_l = MLP(LN(\boldsymbol{z}_l^{'}))\boldsymbol{z}_l^{'} \quad l = 1, ..., L \tag{4}$$

4.4 InceptionV3

Inception V3[16] is a very popular mode in the prediction task. The main idea of the Inception architecture is to find out how to use dense components to approximate the optimal local sparse nodes. There are several version of inception model, including inception, inceptionv2 and inceptionv3. InceptionV3 is the improved version of inceptionV2. Inception V3 has made two main improvements to Inception V2. First of all, Inception V3 optimizes the structure of the Inception Module. Now Inception Module has more types (there are three different structures: 35 × 35, 17 × 17 and 8 × 8).Secondly, Inception V3 also introduced the method of splitting a larger two-dimensional convolution into two smaller one-dimensional convolutions. For example, 7×7 convolution can be split into 1×7 convolution and 7×l convolution. This is called Factorization into small convolutions thought. This asymmetric convolutional structure splitting can handle more and richer spatial features and increase feature diversity. It can be better than symmetrical convolutional structure splitting, while reducing calculation amount.There are totally 295,683 trainable parameter in the model.

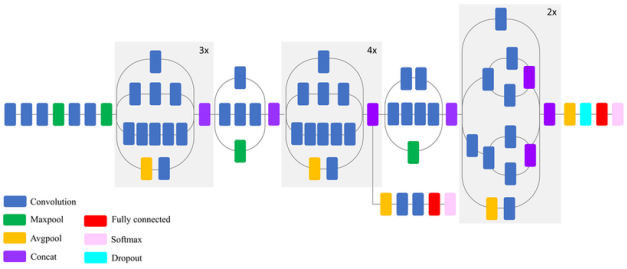

Fig. 5: Model Structure for InceptionV3

5 Experiments

For testing metric, the *classification_report* from sklearn metric module is used. It contain 4 metrics including precision, recall, F score and support. precision is tp/(tp+fp), recall is tp/(tp+fn), F1-score is weighted harmonic mean of precision and recall, support is the number of y_true in each class.tp, fp, fn represent true positive, false positive and false negative.
Random Forest: When it comes to the Random Forest, we use the sklearn to do this. The max depth of the forest is set to 9 and the random state is set to 3.
ResNet50: Before the application of the pre-trained model, pre-processing steps are made. The preview of applying those steps to one image is shown in Fig. 6. After that, we use pre-trained ResNet50 in Keras to extract 2048 dimentional feature vectors. The features are then applied to fully-connected layers to train the classifier with batch size of 32 and epochs of 10.
ViT: First, the pre-trained model "google/vit-base-patch16-224" is loaded. Two linear layer and a dropout layer is applied after the ViT model for output processing. Two linear models have input dimension 768 and 256 and output dimension 256

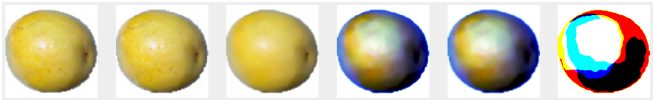

Fig. 6: Visualization of image pre-processing steps. The images from left to right are presenting in the order of **a.**Original Image **b.**Image Resized **c.**Image Denoised **d.**Image equalized and autocontrasted **e.**Gaussian Blur **f.**ResNet50 Preprocessing

and 131. The optimizer is Adam with learning rate 0.0005, loss is cross entropy loss and accuracy is the proportion of correct predicted labels. Batch size is set to be 512 considering the computation power and train for only 3 epochs, with 133 batches each epoch.

InceptionV3: With InceptionV3 function in keras package, we can directly use this function without implementing the whole model from scratch. Due to the limitation of GPU resource, we use the pretrained parameter from ImageNet and continue to train the model based on our fruit360 dataset.And we have trained about 10 epochs on our dataset with the batch size of 32.

6 Result & Discussion

The result is shown in figure 7 and table 1. Figure 7 shows the average training accuracy of training InceptionV3 and ResNet50 for 10 epochs, ViT for 3 epochs and random forest. Table 1 shows the training and testing accuracy from the last epoch and the precision and F1 score for testing data.

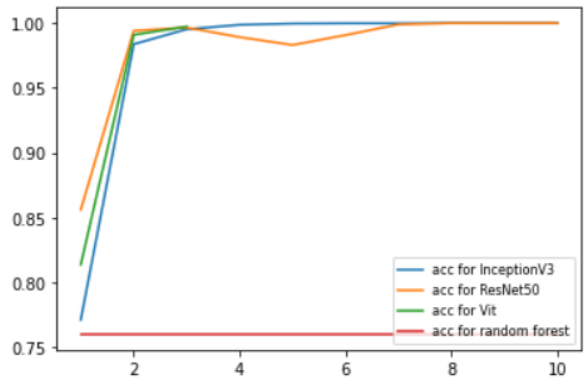

Fig. 7: Accuracy plot for all models.

The predictive result of random forest is lower than the other three deep learning models. The reason for its undesirable result is that random forest is very sensitive to the noise on the image, which will make the model overfitting. It is also the reason why random forest get high accuracy on the training set and low on the test set.

Feature extraction with ResNet50 has the highest precision and F1 score among all three models, which is both around 1.0. However, its performance on the training and validate accuracy of the final epoch is not the highest (ranked 2nd behind InceptionV3). Aside from the possible overfitting of the Inception model. This shows that ResNet50 has a high learning rate.

In the InceptionV3 model, the model can obtain pretty high accuracy in the training dataset and validation dataset. The structure of InceptionV3 is explainable and the deep structure makes it catch the important prediction features fully, which contributes to its good performance. But when it comes to the validation set, the accuracy is lower than the other models. Since inceptionV3 has too many layers, in the training set, it is likely that the model performs a little over-fitting.

ViT model shows good performance with precision and F1 score of 0.98 and 0.98 on testing set. The testing and training accuracy is also over 98%. Only 12 out of 131 classes have a f1-score lower than 95%. Note that this training result is obtained within 3 epochs with the pre-trained model. The ViT model used is already pre-trained on large dataset, which contributes to its great performance. Self-attention layer also allow ViT model to better retrieve information throughout the image and thus resulting in a better prediction.

Table 1: Train, validate accuracy, test precision, F1 score

Model	Train Accuracy	Validate Accuracy	precision	F1 score
Random Forest	94.3%	–	0.763	0.76
ResNet50	99.94%	99.88%	1.00	1.00
ViT	99.7%	98.7%	0.98	0.98
InceptionV3	99.99%	99.95%	0.9364	0.94

7 Conclusion

In our project, we use traditional machine learning method (Random Forest) and deep learning methods (ResNet50, ViT and InceptionV3) to make prediction for the vegetable and fruits on the Fruits-360 dataset. The comparison between the performance of the models is made to find the most suitable model for this classification task. The result shows that every model used in this work is capable to classify 131 different species, where the ResNet50 model can reach the precision of 1, which is the highest thus showing its good adaptation in fruits and vegetable classification. We take a guess that it is for the reason that the size of ResNet50 is suitable. On the other hand, the InceptionV3 model performs the worst in precision, presenting a relatively poor architecture despite its large number of layers. What's more, the results get from deep learning methods are better than that from the machine learning methods, which prove that data-driven deep learning methods can capture the most important features of the data to make a more accurate prediction.

8 References

[1] M. Brewer, L. Lang, K. Fujimura, N. Dujmovic, S. Gray, and E. Knaap, "Development of a controlled vocabulary and software application to analyze fruit shape variation in tomato and other plant species," *Plant physiology*, vol. 141, pp. 15–25, 06 2006.

[2] L. F. Santos Pereira, S. Barbon, N. A. Valous, and D. F. Barbin, "Predicting the ripening of papaya fruit with digital imaging and random forests," *Computers and Electronics in Agriculture*, vol. 145, pp. 76–82, 2018. Available at https://www.sciencedirect.com/science/article/pii/S016816991731030X.

[3] A. Jiménez, R. Ceres, and J. Pons, "A survey of computer vision methods for locating fruit on trees," *Transactions of the ASABE*, vol. 43, pp. 1911–1920, 2000.

[4] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," 2021.

[5] F. Femling, A. Olsson, and F. Alonso-Fernandez, "Fruit and vegetable identification using machine learning for retail applications," in *2018 14th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pp. 9–15, 2018.

[6] H. Mureșan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, pp. 26–42, 06 2018.

[7] M. Oltean and H. Mureșan, "Fruits 360 dataset on github @ONLINE," Sept. 2020. Available at https://github.com/Horea94/Fruit-Images-Dataset.

[8] M. Oltean and H. Mureșan, "Fruits 360 dataset on kaggle @ONLINE," Sept. 2020. Available at https://www.kaggle.com/moltean/fruits.

[9] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[13] A. Mahmood, A. Giraldo, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, R. Fisher, and G. Kendrick, "Automatic hierarchical classification of kelps using deep residual features," *Sensors*, vol. 20, p. 447, 01 2020.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[16] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sensing*, vol. 10, p. 1119, 07 2018.

9 Contribution

The personal contribution in this group project is listed below

- Chong He

  She worked on the coding and analysis of feature extraction and ResNet50. Paper writing

- Mingen Li

  He worked on the coding and analysis of PCA and ViT model. Paper writing

- Entong Su

  She worked on the coding and analysis of random forest and inceptionV3 for prediction tasks. Paper writing