Progress Report (December - February)

Brief information about the dataset

The dataset is the same as the one I have collected using Stellarium. I have discussed that in the previous progress report. Brief information is that It consists of input images and an output target dataset that shows the RA, DEC, and Theta position of the stars for each image. Here is an example that I gave in the previous report:

Figure 1: One of the input sky image

Table 1. Position of the above input sky image

	RA	DEC	Theta	FOV
Input image	55.275°	4.516°	229.248°	70.773

Vision Transformer (VIT) model

As you suggested, I have tested the VIT model [1] for my model. To apply the VIT classifier, I have used the vit_keras in Python which is a package that implements the VIT model. There were several models inside the package for transfer learning such as L/16, B/16 R50+ViT-B_16, etc. The one that I have used is L/16. Because it has the highest accuracy on the cifar10 dataset which is an accuracy of 99.36%. Here is my code to implement the L/16 VIT model.

```
base_model = vit.vit_l16(image_size = (512,512), activation = "sigmoid",
include_top = False, pretrained=True, pretrained_top=False,
weights='imagenet21k+imagenet2012' )
```

After uploading the model, the first step was to freeze the layers of the L/16. The reason is that it would take so much time to train it again, and it was already trained with the imagenet21k and imagenet2012 datasets.

```
for layer in model.layers:
    layer.trainable = False
```

Then I added a few layers to make it appropriate for my problem. First I have added the dense ReLu layer with 256 nodes. Then output layer with 3 nodes with a sigmoid activation function was added which corresponds to each variable's Right Ascension, Declination, and Theta.

```
x = base_model.output
x = Dense(256, activation='relu')(x)
output_layer = Dense(3, activation='sigmoid')(x)
```

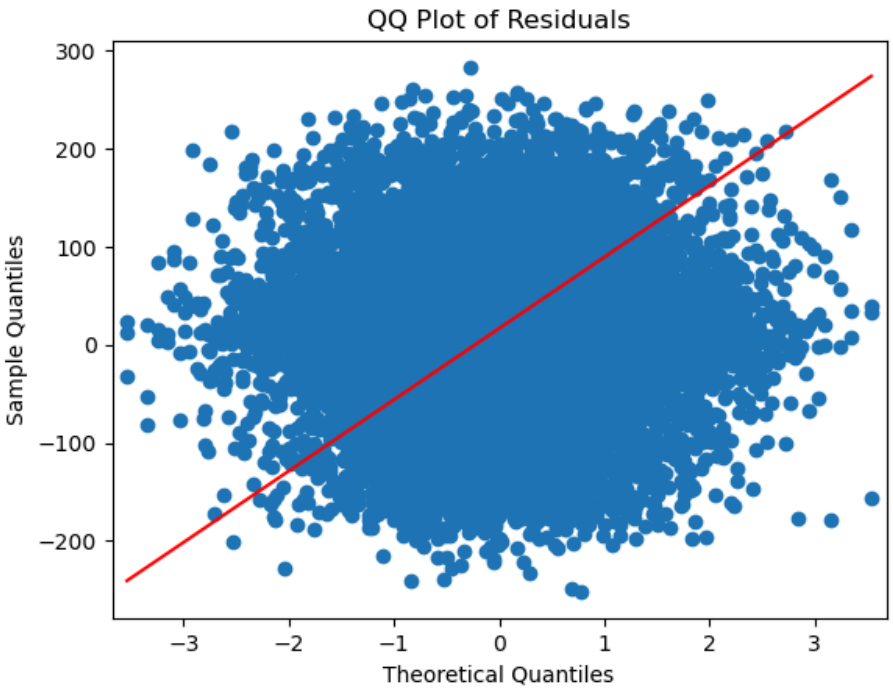
Then the model was defined with inputs as the base model which is L/16 and output as the output layer that I have created. The model was compiled using an “Adam” optimizer with a mean-squared-error (mse) loss function. The callback was defined so that if model accuracy is higher than previous epochs, the model will be saved on a hard disk.

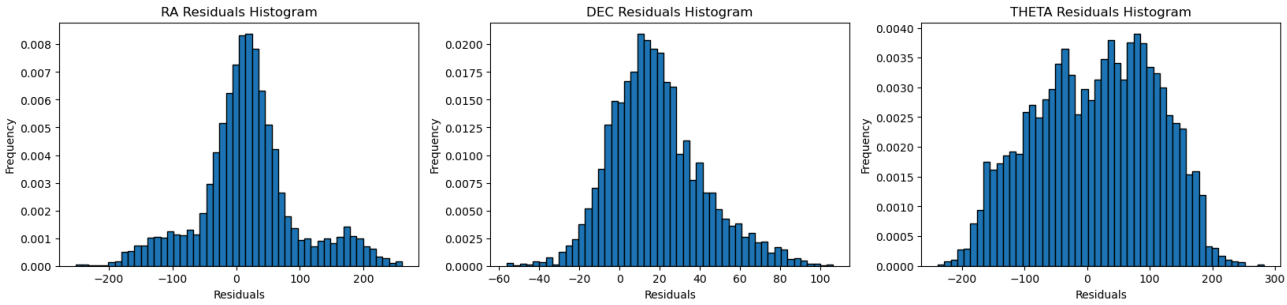
In the first stage, the model was defined to train for 100 epochs with the bath size 32. However, every epoch took about 8 hours despite of freezing layers of the L/16 and continued for several weeks. After several epochs, the model was terminated because of memory and CPU limitations of my laptop regardless of its high computational parameters such as i9-12 CPU, and 3070Ti GPU. Even this laptop wouldn’t be able to train the model for 100 epochs. After the termination of the model, the model was tested with the test set. Here are the metrics of the model after calculated using the test set which took about 3 hours for prediction:

Table 2: Metrics of the model

	RA	DEC	Theta
MAE	57.024	22.624	82.849
MSE	6268.532	869.441	9526.947
RMSE	79.174	29.486	97.606
R2	0.434	0.421	0.113

I have also visualized the residuals as scatterplot and histogram to see the spread of the error, and accuracy is printed with a certain uncertainty threshold:






```
RA accuracy with 5-degree uncertainty =  7.371794871794872
DEC accuracy with 5-degree uncertainty =  15.144230769230768
THETA accuracy with 5-degree uncertainty =  2.864583333333333
```

As you can see the result of the model is not good, and it is not possible to use as the star tracker algorithm. The reason, in my opinion, after researching VIT, VIT classification wouldn't be successful on this task even if I had trained it for 100 epochs. The reason is that this kind of model is more suitable for classification and gives better results on classification problems. I couldn't find the resource that implements the VIT classifier for regression problems.

ResNet50

After the research of several models, The ResNet-50 model was selected as the most suitable option with the decision being substantiated by various factors including its ability to effectively handle complex features and mitigate issues related to vanishing gradients owing to its deep residual structure[2]. Right now, I am training this model for several epochs to see the performance of my problem. What can I say is that, after the baseline model, it gave a better result than the VIT classifier. I will give you the progress report of the ResNet50 model in the next progress report.

Research papers

[1] A.Steiner, A. Kolesnikov, X. Zhai, etc. “How to train your ViT? Data, Augmentation, and Regularization in Vision Transformers”

[2] K.He, X.Zhang, S.Ren, and J.Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016, pp. 770-778.