EE 519: Deep Learning Theory & Fundamentals **Spring 2023**

Mini Project 2

Due: May 21, 2023, 11:59PM PT

Student Name: Sandy Dash *Instructor Name: John Lipor*

1 Problem Description

The goal of this project is to develop a predictive model using Convolutional Neural Network (CNN) which studies Favorable Structural Settings (FSS) in the images of Detrended Elevation Maps (DEM) and predicts heatflow residuals. Accurate prediction of heatflow residuals also known as geothermal favorability, is of interest to United States Geological Survey (USGS) so that earth's heat can be used to produce electricity.

We are given a dataset of 222 DEM patches of size 40 km x 40 km which equals to 200 x 200 pixels, with 222 labels which are heatflow residuals to train the CNN. We are also provided with 56 DEM patches as test dataset without labels. The training dataset is further split into training and validation sets using the 80:20% ratio.

Dataset	**Images**
Train	177
Validation	45
Test	56

2 Exploratory Data Analysis (EDA)

To explore the image data, I first plotted all of 222 training images with their labels and indexes. Also, I plotted the test images to compare it with training dataset. Insights about dataset is mentioned in figure caption. As part of filtering the data, I removed some of the images which has very little information or striations. I did this, to see if the model provides consistent prediction. In the following figures, images in red boxes have been deleted from the dataset.

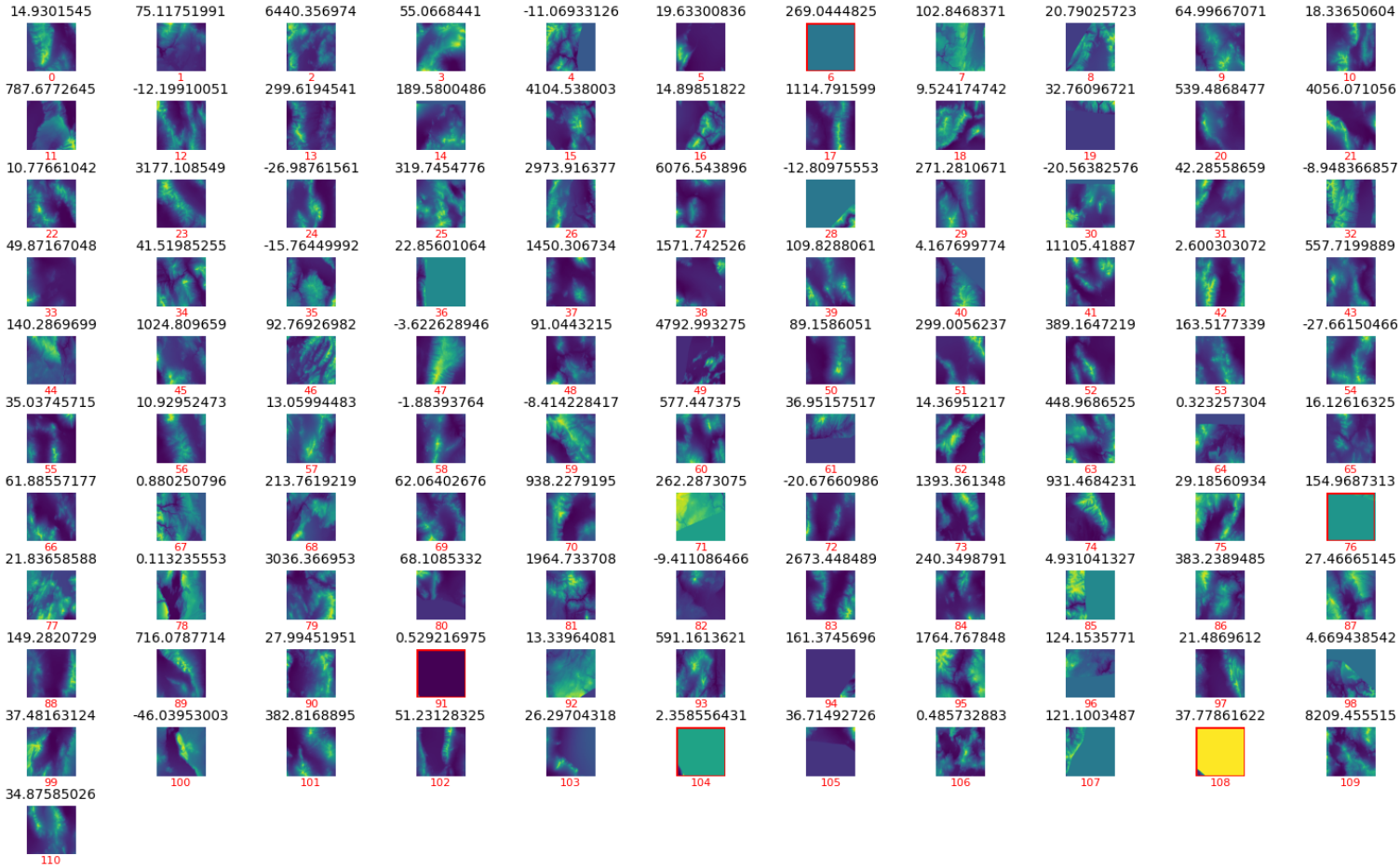

Figure 1: Training dataset of first 111 DEM patches. Numbers on the top of the images are labels. Numbers in red on the bottom of the image are indexes of the image. At first I was trying to find if there is any correlation between a higher label and striations seen on the image. Lets consider index 5, it looks like the striations are only on 1/3rd of the image and rest of the patch looks empty. If we consider that as a baseline and say label 19.6 looks like this, then as we move to index 6, it looks like the whole patch is empty yet label is 269. Similarly index 2 has a label 6440 which makes sense because there are lots of features in the image but then how come index 25 is labelled as 22.8? Therefore I concluded that may be a human eye cannot find a correlation between the labels and the patches.Images with **index 6, 76, 91, 104, 108 are deleted**.

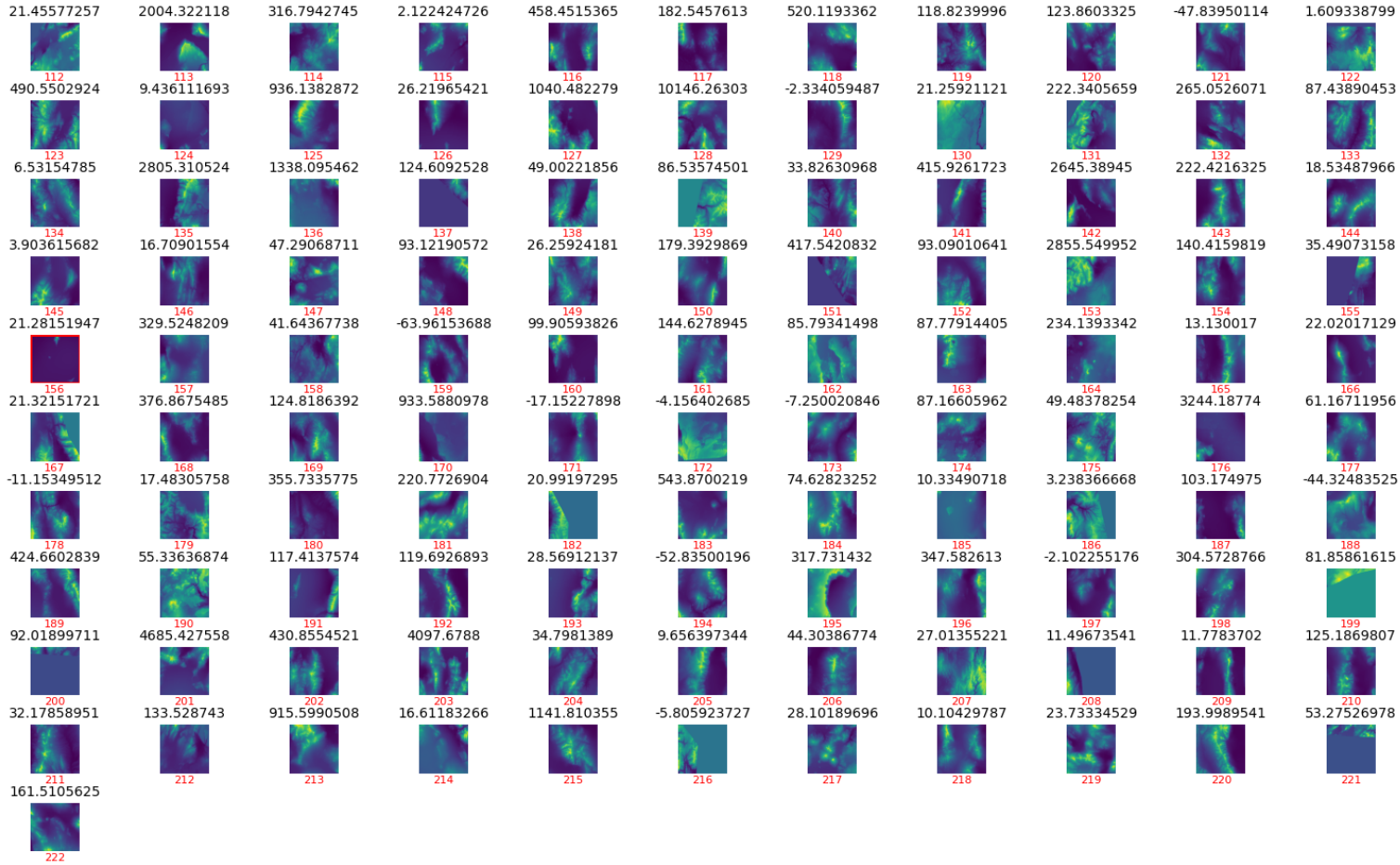

Figure 2: Training dataset of remaining 111 DEM patches. **Image with index 156 is deleted**. I could have removed indexes 137, 199, 200 and 221 as well but that would mean 4.5% data reduction as opposed to 2.7%. Since this is a small training dataset, I chose to have more training data at the cost of inconsistent prediction.

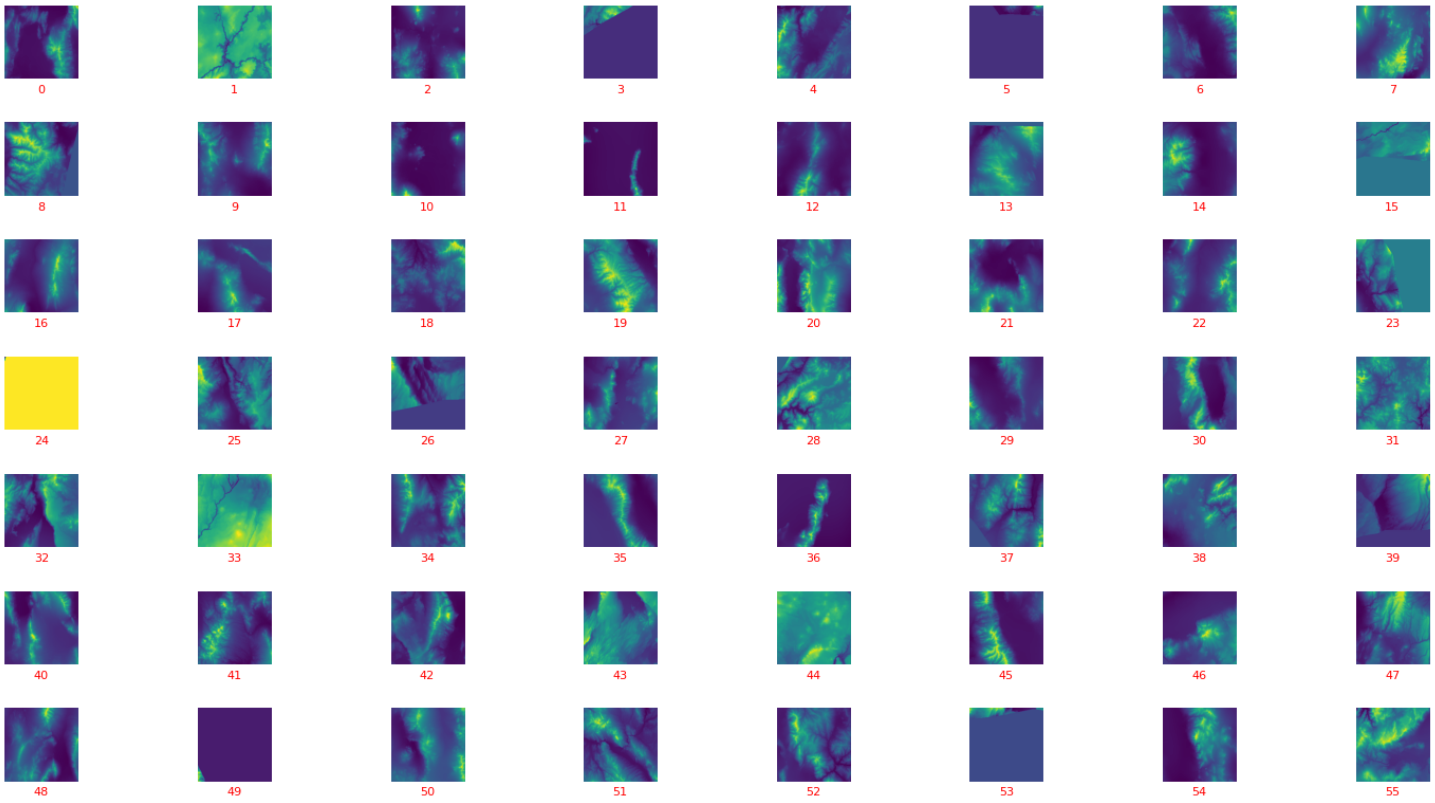

Figure 3: Test dataset of 56 images. Here I think patches with index 3, 5, 24, 49 and 53 have very little striations which is about 9% of the test dataset. This was another reason I chose to not clean up the training set entirely by removing all of the patches with minimum striations.

I chose two approaches for this problem as stated below:

1. **Ordinal classification problem**: In this approach the CNN model is presented with training images and ordinal ranks as labels instead of actual heat residual numbers. This is fine for classification because here the CNN will predict the probability of each ordinal class for a given image. So the prediction does not lose its meaning.

2. **Regression problem**: In this approach I use CNN to study the training images and learn the labels. But because we care less about the actual heat residual and more about the fact that the image belongs to which ordinal class, therefore initially I thought the label should be the ordinal classes. But when I looked at the floating point predictions such as 2.5 for lets say label of 3, then what does the prediction mean? If we round it up then the prediction will match the label, but what if the label was 2? In either

case 2.5 is a good prediction for a label of 2 or 3. **Predictions lose its meaning if the problem is considered as a ordinal regression problem**. Therefore for this approach, I chose labels as the heat residual numbers. After the model has made the prediction then bin and rank the predictions and labels. The range of heat residual is [-63.96, 11105.41] and its not a normal distribution. Because we care about ordinal rank of the prediction, I chose to cap the residuals using the following logic:

```
if heat_resid < 0 :
   heat_resid = 0
elif heat_resid > 300 :
   heat_resid = 300
```

This logic helps to maintain mean at the center of distribution.

3 Approach

As stated before, I have considered two approaches to solve this problem: 1) Ordinal classification and 2) Regression. Here I will describe both the approaches in detail.

Ordinal Classification Approach

Since the dataset is so small, I decided to use the same AlexNet that I built in HW2. The network has

Types of Layers	**Number of Layers**
Convolution	5
BatchNorm	5
ReLU	7
MaxPool	3
DropOut	2
Linear	3
AveragePool	1

For faster training and to increase the stability of the output, I have chosen to keep BatchNorm layers and to prevent overtraining I have kept the DropOut layers. Since its a classfication approach therefore the loss function is CrossEntropy. For gradient descend or optimizer, I had used Stochastic Gradient Descend (SGD) for HW2 but for this project that was **causing exploding gradient issue** for both classification and regression approaches. Therefore, I chose Adams optimizer for both classification and regression approaches. To prevent the issue of reducing the pixels to zero, I have kept the kernel size as 3x3. I didn't change this at all throughout the experimentation of this project. Bigger kernels can help with faster feature extraction given the higher resolution of the images but the datasize is so small that training time was not a concern on my workstation.

Regression Approach

This approach as described in the EDA section. It also uses the same CNN described in Ordinal Classification approach. The only differences are :
Loss function - HuberLoss
CNN output classes - 1

Metric for Evaluation of the Model

The model performance is measured by Mean Absolute Error (MAE), also known as L1 loss, of validation set as described below:

$$\ell(y, \hat{y}) = (|y - \hat{y}|)/Validation\_size, \tag{1}$$

where $y, \hat{y} \in \{1, \cdots, 4\}$.

Data Augmentation

As described in the problem description, the training dataset comprises of only 177 images, therefore it was important to add different data transformations to increase the total number of the images that the model gets to learn from. I studied about image augmentation choices and their impact here. I chose a **baseline transformation which comprises of Horizontal Flip, Vertical Flip and RandomRotate between 0 degrees to 359 degrees**. Then I added one new data transformation techniques at a time to see the change in performance. The table below shows MAE of validation set for **Monte Carlo Cross Validation** over 10 iterations.

Types of Transformation	**MAE on Validation set**
No transformation	1.4
Baseline(B)	1.34
B + RandomPerspective (RP)	1.42
B + GaussianBlur (GB)	1.44
B + Gaussian Noise (GN)	1.47
B + RP + GB	1.29
B + RP + GB + GN	1.49

From the above table it is clear that the model performance for the classification approach only depends on two things:

1. **Random split** - As described in the figure captions of the EDA section, despite deleting six images from the dataset, it still contains alot of images which has striations on only one side of the image and the rest of the image looks empty. So lower MAE is really a matter of chance, if the validation set has images with more striations then the model performs better.

2. **What features does the model choose to focus on** - At first I was not sure about this factor but when I looked at the Class Activation Maps (CAM), I saw that the CNN is also focusing on finer details in the images which do not look like striations. Check figure captions for figures 8 and 9.

Therefore I chose just the baseline transformations for the model.

Hyperparameters

I used optuna to find the best values for the following hyperparameters but because of the reasons stated above, which determines the model performance, these hyperparam values did not provide consistent MAE values. Therefore I **chose the values that executes faster while providing low MAE values**. I got these values by using the trial and error method. Learning rate is two orders of magnitude higher than what's suggested by optuna because I am using Batch Normalization which allows for higher learning rate. Keeping the epoch low helped with preventing overtraining which is proven in the plots of training nd validation loss. Check figures 4 and 5 Number of epochs for classification and regression approaches are different. The table below summarizes all of the hyperparam values suggested by optuna and what I chose.

Hyperparam	**Optuna value**	**Chosen value**
epochs	98	10(C) 66(R)
batch size train	40	20
batch size val	27	45
learning rate	0.00101	0.1

Validation set does not need to have smaller batches because backpropogation is only happening for training.

4 Challenges

Many of the technical challenges of using CNN models and CNN visualization techniques were already explored and documented in HW2. HW2 prepared me well for this project especially because this is my first exposure to image data and CNNs. Other challenges are described below:

1. **Small dataset with many images with very few FSS.**
2. **Only one feature that is FSS to learn and predict geothermal favorability based on that as opposed to 28 features in the tabular data of MP1.** - More features helps the model to learn and perform better.
3. **Useful features are not located at the center of the image** - This prohibited us to use RandomCrop or padding data transformations because we might crop or zoom out useful features accidentally.
4. **CUDA error:device-side assert triggered** - I have hit this error several times during this project for the classification approach. This link helped me resolve this issue.
5. **DataType error on targets** - To apply the data transformation I wrote a class where I needed to specify target datatype. For classification approach it should be LongTensor whereas for regression approach it should be FloatTensor.
6. **Exploding gradient problem** due to SGD optimizer. I chose Adams optimizer for both regression and classification approaches because SGD lacks adaptive learning rate and second-order optimization which provides more balance to parameter updates.

5 Evaluation

In this section I will provide model performance for both classification and regression approaches. I will also provide figures from **CNN visualization for the classification approach** and write conclusions about those figures in their captions. Unfortunately I had troubles saving the trained parameters for the best performing model therefore the visualization images are captured at slightly different model performances which are mentioned in figure captions.

Best Model Performance

Following tables show the best model performances for validation set, achieved using both the approaches. These are collected by performing **Monte Carlo Cross Validation** over 10 iterations.

1. **Classification approach**

Performance metric	**Value**
Ordinal classification accuracy	45.5%
MAE	0.98

2. **Regression approach**

Performance metric	**Value**
Ordinal classification accuracy	36.36%
MAE	1.05

From this we can conclude that **classification approach is better** for this problem because not only it shows better performance but also the model generalizes very quickly that is in 10 epochs as opposed to 66 epochs for regression. The inconsistency of the result is an issue for both approaches and probably can be fixed by doing the following:

1. Removing all of the images from training and validation set which has very little FSS. But we cannot remove such images from the test set so this is not an option.
2. Either using co-ordinates of features and feeding that into the model or carefully cropping the images to include only the useful information.

Plot for Training vs. Validation Loss

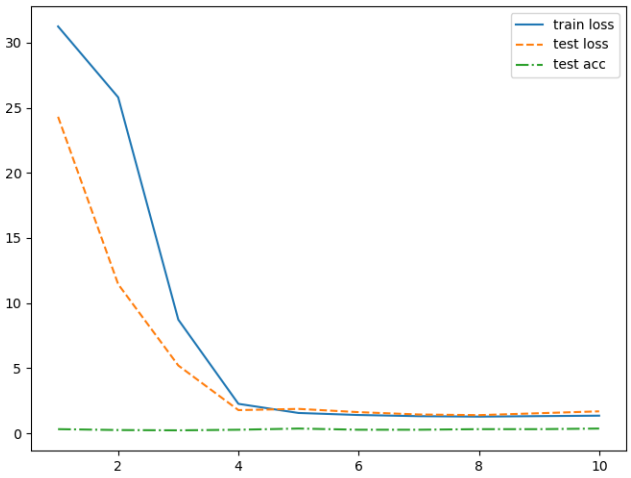

Figure 4: Training vs. Validation Loss for classification approach captured at MAE 1.36. The plot shows the expected downward trend for both training and validation losses as epoch increases. Also since the validation loss is not rising again, it shows that the model is not overtraining.

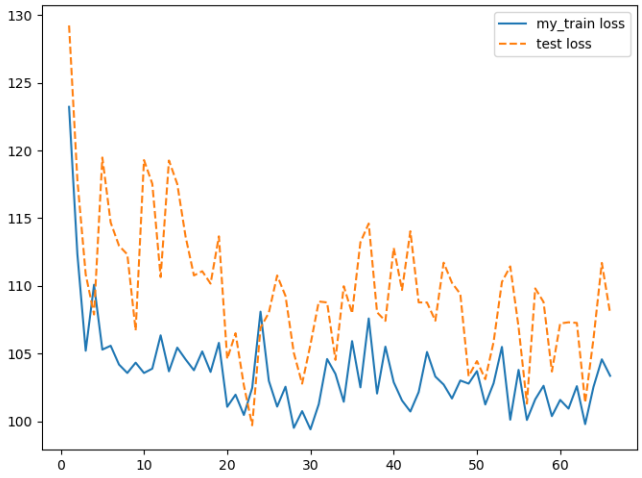

Figure 5: Training vs. Validation Loss for regression approach captured at MAE 1.05. The model has more variation in errors from epoch to epoch.

Feature Maps

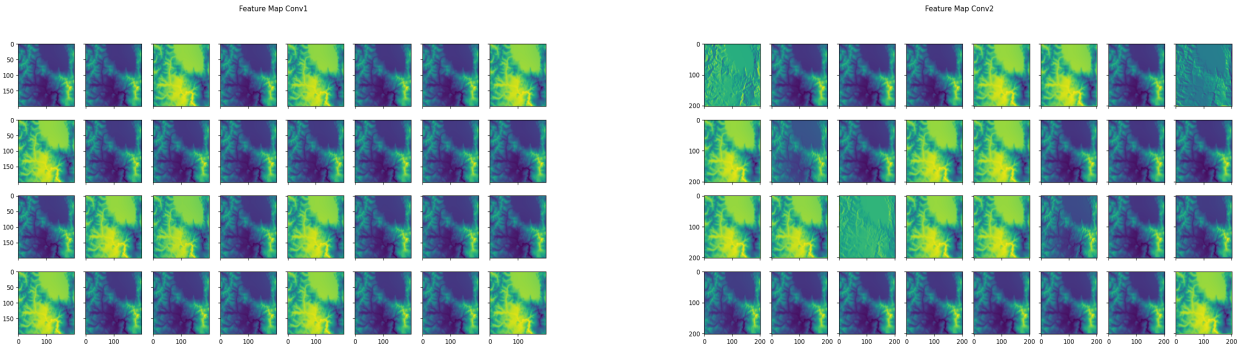

Figure 6: Feature maps from the first two convolution layers of AlexNet. In this image the number of FSS are more and the clarity of the striations shows that the model is able to learn the useful features.

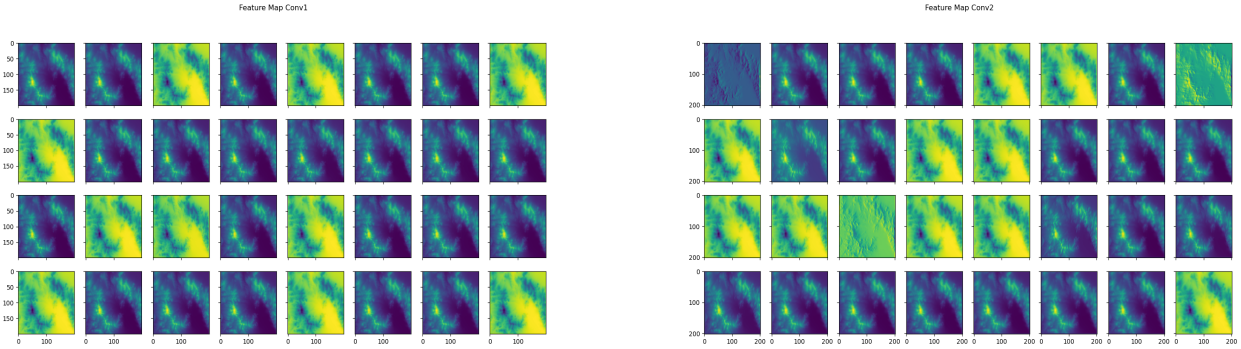

Figure 7: Feature maps from the first two convolution layers of AlexNet. In this image the number of FSS are fewer than figure 4 but the model is still able to identify, learn and focus on the striations.

Class Activation Maps (CAM)

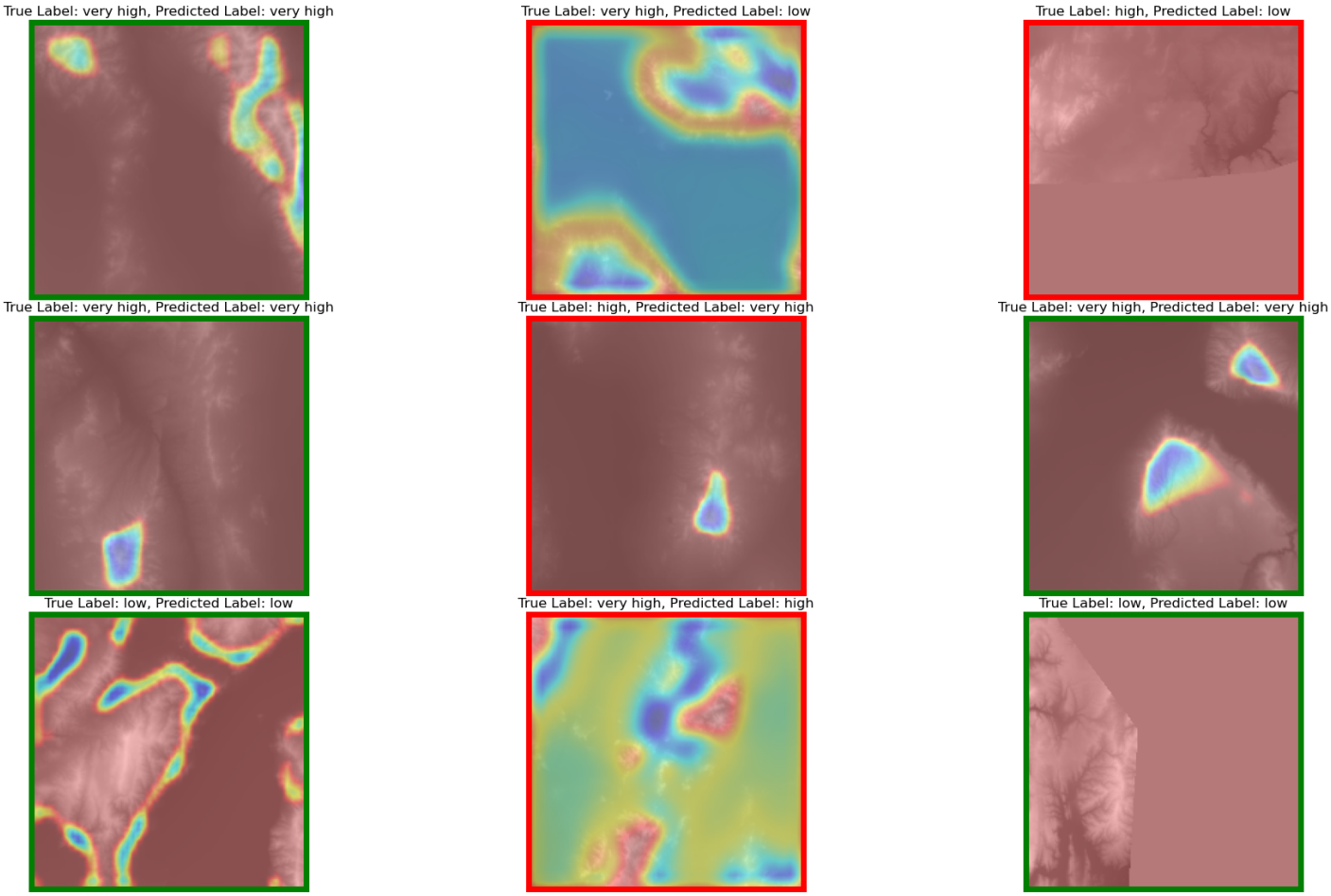

Figure 8: CAM image captured at model classification MAE 1.09. The green borders indicate the model has predicted correctly. The interesting thing to note is that the model is able to find less important sections within FSS e.g. the top left and the first subplot of second row. But on the other hand the model also focuses on all of the empty spaces which are red in the top right and bottom right subplots.

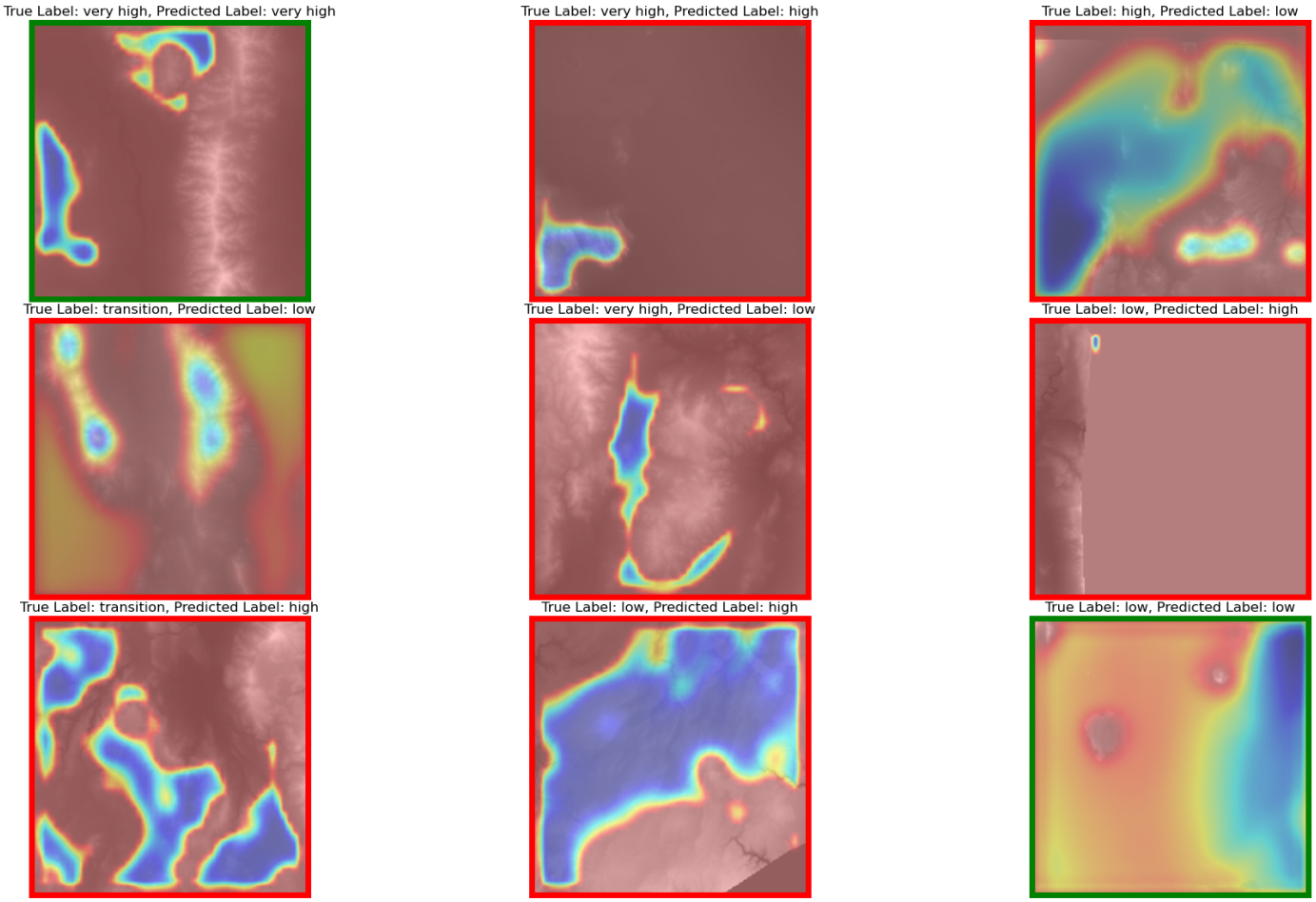

Figure 9: CAM image captured at model classification MAE 1.43. In this image we can see that the model is not performing well. The top left subplot is mostly red, so the model thinks that there are alot of useful features and hence it labels it as very high. But the center image has so many FSS and the model thinks that it should be labelled as a region with very high geothermal activity but the label says low. This is where I think the human and the model fail to understand the correlations between the amount of FSS and its corresponding label. Another evidence of the model not being able learn properly is shown in the top right subplot. Most of the image does not have useful feature as indicated by the blue and green CAM colors but the model predicts high.

Guided Back Propagation

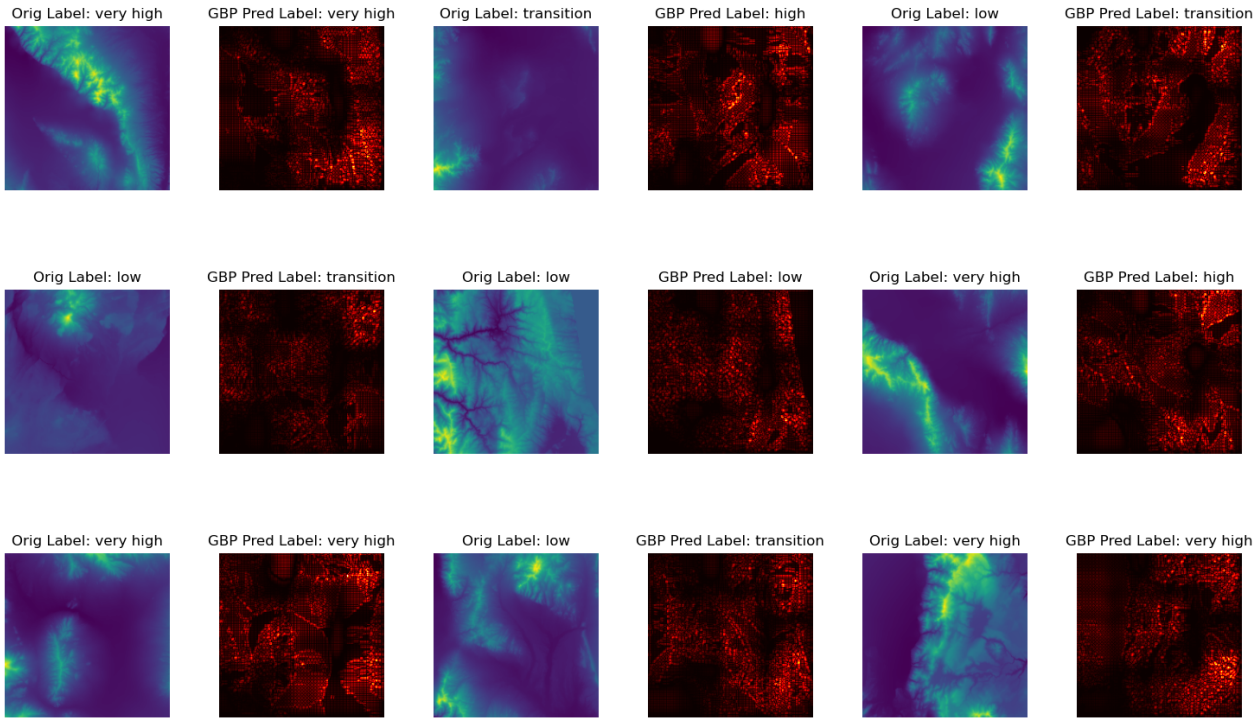

Figure 10: Guided Back Propagation (GBP) image captured at model classification MAE 0.95. Four out of nine predictions are correct. One thing that really captured my attention was the subplot of last row, middle column. If you zoom in you will see some very light patterns in the blue region which are not easily visible to the human eye but I only found out about those by looking at the GBP image. I think that the prediction is wrong because the model emphasizes on those light patterns and thinks that there are more FSS. But then the model does the same thing for the bottom left image and the prediction is correct. This brings us back to the same question that I have stated in figure 9 and in the EDA figures that the correlation of amount of FSS w.r.t to their labels is not understood.

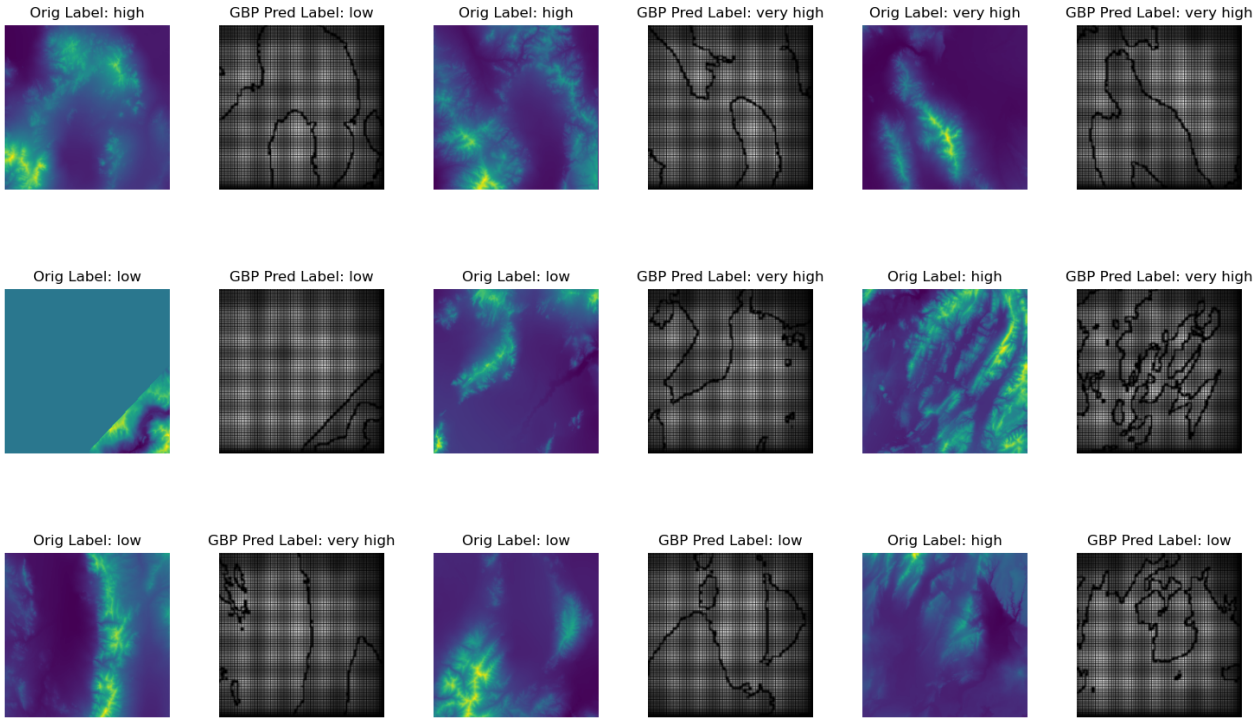

Figure 11: GBP captured at model classification MAE 1.39. Three out of nine predictions are correct. The reason included this picture is because I have tried three different options for cmap and I think this one clearly shows the features that the model is focusing on by outline it in black. However I could not find a way to remove the matrix look in the background.

6 What I learned

I had spend alot of time learning all the tools, libraries, CNN network types, their usage in different types of problems and visualization techniques for HW2. So thankfully I did not have to deal with anything new. However there were a few new errors that popped up for this project which I have described in the Challenges section.

7 Conclusion

In this project we were given a task for predicting geothermal favorability using patches of Detrended Elevation Maps (DEM). We were required to build a predictive model using Convolutional Neural Network (CNN). I chose to solve this problem using two approaches:

1. Ordinal Classification Approach
2. Regression Approach

while using the same CNN for both the approaches. Based on the model's performance, time taken to generalize and speed of execution, classification approach is better than regression. Despite all the efforts and experiments, the model could not be tweaked to provide consistent performance with the chosen training dataset. However if the dataset is filtered further to keep only the images which has more FSS then the model performance can be consistent but there will be a big penalty on the model performance for the test images because 9% of the test dataset has images with less FSS. Based on the CAM and GBP images, it can also be deduced that for this dataset CNN might perform better if features are concentrated at one location than being spread out throughout the 200 x 200 pixels.