Segmentation of Crop Damage

1st Arpad Voros
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
aavoros@ncsu.edu

2nd Josh Beerel
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
jbeerel@ncsu.edu

3rd Zackery Miller
Electrical and Computer Engineering
North Carolina State University
Raleigh NC, USA
zamiller@ncsu.edu

***Abstract*—This paper presents a deep learning model that processes images of crop fields to assess their health status with the key objective of identifying which areas of the fields have been damaged. Image data of crop fields was downsampled due and used as the training data, employing semantic segmentation methods to define 'damaged' and 'not damaged' sections of the input images. Mean IOU was the key metric used to determine the performance of the deep learning model and enabled an easy comparison of these models. This paper also discusses future work that could further enhance the results such as the current model structure sticking at local minima.**

***Index Terms*—CNN, computer vision, semantic segmentation, classification, agriculture, crop damage**

I. Methodology

Crop images were provided for the training of our models; however, these images required a fair amount of pre-processing before they could be loaded into our deep learning models. The Colab Environment used in this project has a maximum RAM allocation and with the full-size images being 3-channel RGB at a resolution of 3456×4608 pixels, it required that the images be downsized. Fortunately, this pre-processing step allowed for faster training of the models as a result of the smaller image size. The images were initially downsized in MATLAB before being imported into the Colab Environment and the following sizes were considered:

Resolution (%)	**Resolution (px)**
2.5%	96 × 128
5.0%	176 × 240
5.3%	192 × 256
10.0%	352 × 464
20.0%	704 × 928

The downsampling process was not ideal as the U-Net architecture [1] used in this project requires the pixels along each axis to be divisible by 2^N, where N is the number of down/upsampling stages. The model we employed used $N = 4$ [2], thus each dimension was required to be a factor of 16 and this was accomplished using Eq. 1, where size is a 2×1 matrix of height and width, size$_{down}$ is a 2×1 matrix of the downsampled height and width, and scale is the % of scale factor selected from the table above.

$$\text{size}_{down} = 2^N \left\lceil \frac{\text{size} \times \text{scale}}{2^N} \right\rceil \tag{1}$$

Another key consideration in the data was that three train/annotation pairs were unable to be considered due to poor image quality that resulted in too much blur or high exposure that produced near-white blank images. With all of the images edited in MATLAB, they were imported into Colab where the following process was employed:

1) Training images were downscaled by 255 to get a floating point value for each datapoint (pixel)
2) Annotations/Labels were downscaled by 255 in a similar process as step 1, and averaged along all three channels
3) Annotations/Labels were reshaped to (height, width, 1) since the channels were averaged to 1-channel from point-of-view of the data
4) Floating point values were finally rounded to 0 or 1, generating a simple binary mask

This procedure allows the training images to retain their image characteristics while applying this binary mask. The purpose behind the implementation of a binary mask is due to the structure of the data: crop segments are determined as either damaged (1) or not damaged (0). This was fed into a U-net architecture [1] with the structure shown in Fig. 1.

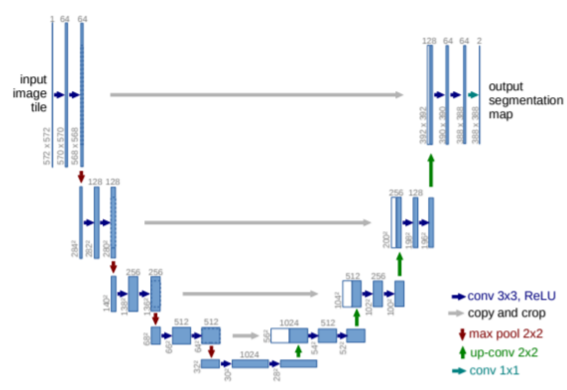

Fig. 1: Model Structure of U-Net [1]

The training metrics for this model structure are defined in Table 1.

Three evaluation metrics were employed as binary accuracy provided good insights for the training stage, mean IOU was used for the testing stage, and standard accuracy metrics was a decent metric for both. The structure used consists of blocks as follows: b_1 = convolution $\rightarrow$ dropout and b_2 = convolution $\rightarrow$

TABLE I: Model Training Metrics

Parameter	Value
Epochs	20
Evaluation	Binary Accuracy Accuracy Mean IOU
Optimizer	Adam
Loss Function	Cross-Entropy

max pooling. These two blocks were alternated N times before the layers were concatenated at the end.

II. Model Implementation and Results

The following Fig. 2 shows the initial image of the crop field and the processed image used for training the model. The white bars indicate areas of the crops that are damaged while the black components are the undamaged crop fields.

(a) Image of Crop Field before processing (b) Processed Image used as training input

Fig. 2: Images for Model Training

Training the aforementioned model for the specified twenty epochs yielded impressive results for the binary accuracy. In this metric, the binary training accuracy was 96.37% with a test accuracy of 92.3%; however, comparing this with the mean IOU indicated performance along this metric was not that impressive. For mean IOU, the training accuracy was 47% and testing accuracy was 43%. Attempts to achieve a greater accuracy with this metric did not yield positive results, with all of the resultant attempts have sub-50% accuracy. The standard accuracy metric, in this regard, does not matter as the binary accuracy metric is significantly better with respect to model performance. Fig. 3 below shows an example output prediction from the trained model.

The white pixels in Fig. 3 indicate sections of the image that were classified as damaged, while the black indicate otherwise. Comparing this image to Fig. 2b shows that the model employed does not do an effective job at fully categorizing the segments that are determined to be damaged. This issue with the model was found to be due to the Convolutional Structure dropping the horizontal features of the image while retaining the vertical features, which meant that model began mistaking horizontal damage for vertical damage. This was further compounded in the concatenation stage of the model which led to a much greater emphasis on vertical shape as

Fig. 3: Sample Image for Damage Prediction

opposed to the horizontal shape. This can be seen in Fig. 4 and Fig. 5 which compares the image at the layer before the convolution and how it looks after [3].

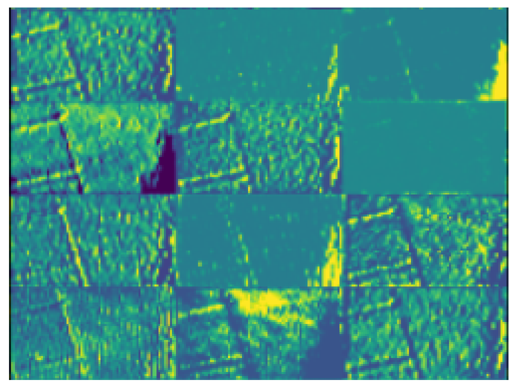

Fig. 4: Convolutional Layer with Horizontal Features

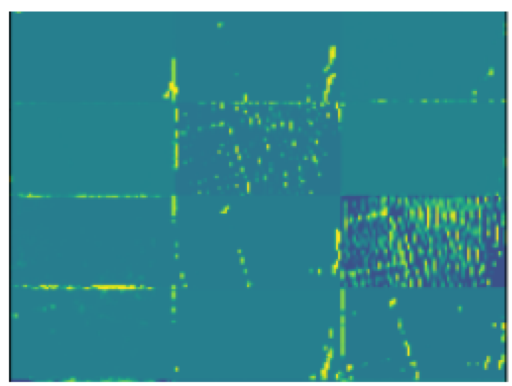

Fig. 5: Convolutional Layer without Horizontal Features

The difference between these two figures is obvious as Fig. 4 still has remnants of the crop field in its structure while Fig. 5, in some cases, produces a blank image by removing those features. This leads to the development of the prediction seen in Fig. 3 that is heavily dominated by the thin line of crop damage, which is identifiable in Fig. 2a. The reason for this will be discussed in the following Evaluation section.

III. EVALUATION

As discussed in the previous section, there were a few key issues from the model trained for this project. The model's convolutional network resulted in the loss of horizontal features in the prediction stage. This key limitation is suspected to be do the network reaching a sticking point at a local minima that preferentially filters for vertical features in the crop field. This is also seen when rotations are applied to the data as shown in Fig. 6.

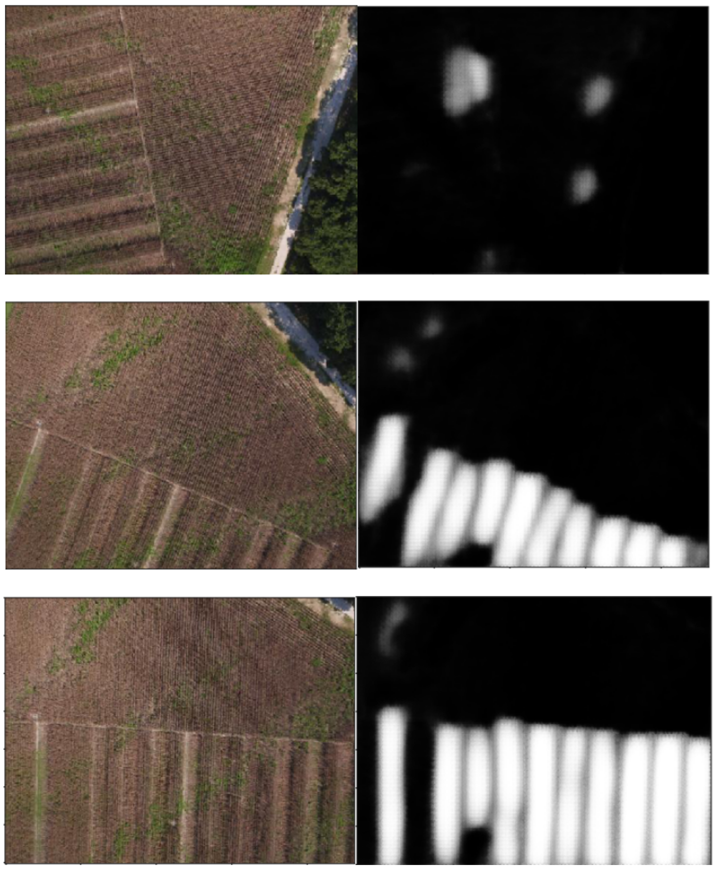

Fig. 6: Rotated Crop Images

Fig. 6 demonstrates rotational angles of the crop field. This figure demonstrates the bias of the model toward vertical features in the structures. As the image of the crop field becomes more vertical, the accuracy of the prediction increases substantially, with the last image capturing almost all of the damaged crop segments.

A. Future Work

Future work will need to address the key issue of bias in the prediction due to the erasure of the horizontal features. One way this can be achieved is through the generation of data that includes more horizontal patches of damaged cropland. Ideally this step alone would improve the accuracy of the model, as it would not require specific orientations of the images to fully capture the extent of the damage.

Further research could be conducted into the design of architectures that suite the semantic segmentation required classification of crop damage. In our implementation, the U-net provided promising results but was limited by the filtering of the CNN layers. A more in-depth analysis into other viable structures could lead to the development of more accurate results. Furthermore, optimization of the model hyper parameters could have received more consideration in this project.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[2] K. Åmdal Sævik, "U-net image segmentation in keras," 2018.

[3] G. Pierobon, "Visualizing intermediate activation in convolutional neural networks with keras," 2018.