Implementation of CNN-Transformer Hybrid on Crop Classification using Remote Sensing Data

David Hwang
hwan0259@umn.edu
University of Minnesota

Abstract

Implementation of model architecture is necessary to ensure that research results are correct and consistent. It also enables people to explore and experiment with the boundaries of the models. However, it can be difficult to replicate the model architecture devised by other researchers as code is either not provided or the implementation details are vague. In this article, I explore the problem of crop classification by implementing CNN-Transformer hybrid architecture created by Zhengtao et al. [1]. This approach uses a convolutional neural network (CNN) approach to spatial-spectral unification of multi-band, multi-sensor data from Sentinel-2 and Landsat-8 to obtain a multi-temporal sequence to mine phenological patterns, the relationship between crop cycles and seasonal variations. The multi-temporal sequence is then passed to a Transformer which is connected to a feed-forward layer and softmax layer for classification on a wide range of crops within central California. In the original study results show that the CNN-Transformer approach yields better overall accuracy over random forest (RF-200), support vector machine (SVM-RBF), Multitemporal CNN, and CNN-LSTM. Implementing my own version of the CNN-Transformer model and replicating the original study with the addition of an ablation study, which takes the CNN-Transformer architecture and removes the transformer module, the results that I obtained show that CNN-Transformer model performed worse than traditional machine learning methods RF-200 and SVM-RBF, and a significant performance improvement was not seen over the ablation model.

1 Introduction

The CNN-Transformer hybrid architecture created by Zheng et al. was chosen for implementation as it merges both CNN and Transformer architectures which are typically used for applications in different fields [1]. CNNs are widely used in the field of computer vision (CV) by extracting and learning features on image data [2] and Transformers are commonly used in natural language processing (NLP) tasks [3]. The fusion of both deep learning architectures is novel in its approach especially with crop classification. The CNN Transformer hybrid architecture also makes usage of spatial-spectral unification (SSU) which enables the architecture to take advantage of dense temporal remote sensing data by combining Sentinel-2 A,B and Landsat-8 by unifying the spatial resolution and spectral bands of the multi-band multisensor images [1]. As a way to see these concepts in action I implement the architecture and replicate the original study on crop classification.

1.1 Crop Classification

The task of crop classification is to accurately map crops growing within a specific region with the goal of monitoring crop health, crop yields, and agricultural management [4]. Crop classification is also able to be done on a very large scale. The US Department of Agriculture (USDA) National Agricultural Statistics Service (NASS) uses crop classification annually to report land cover data nation-wide [4]. To meet the goal of maintaining a cost-effective and regular source of agricultural data, various satellites have been deployed to collect information on the energy that is reflected from Earth [5]. With space-borne remote-sensing (RS) satellites like Sentinel-2A, Sentinel-2B, Landsat-8, and MODIS, crop classification can be done efficiently using remote-sensing images that the satellites capture [5].

As multiple remote-sensing satellites exists it seems possible to combine images provided by different satellites to obtain an enhanced dataset to improve crop classification [6]. However, differences in how satellites collect data can create complications. For example, Sentinel-2, which is composed of satellites Sentinel-2A and Sentinel-

2B, and Landsat-8 both collect multi-spectral data which is the energy that is reflected from the earth in specific wavelength ranges which are referred to as bands. Sentinel-2 collects 13 different bands and Landsat-8 collects 11 bands. So the data collected between the two satellites are not identical. Another complication also exists with the usage of bands. In a remote-sensing image in order to represent the various bands of the satellite there exists different physical dimensions, called the spatial resolution, that represents a pixel of an image to store the spectral data of the band. Sentinel-2 has bands with spatial resolution of 10m, 20m, 60m while Landsat-8 has bands with spatial resolution 15m, 30m, 100m. Satellites also collect data at different rates. Sentinel-2 has a revisit interval, time period between consecutive observations over a location, every 5 days and Landsat-8 has a revisit interval of 16 days [5].

To unify the differences between satellites the CNN-Transformer model makes use of spatial-spectral unification created by Qian Zhang et al. which utilizes CNNs [7]. By combining Sentinel-2 and Landsat-8 remote-sensing data dense the revisit interval can be shortened to 3-5 days [5]. Using dense multitemporal remote sensing data the original study claims that phenological differences are able to be mined with the deep learning architecture which improves performance on classification[1].

1.2 Related Work

One approach to crop classification using remote sensing data is to use traditional machine learning architecture. In 2007, one study used a decision tree algorithm on MODIS imagery to perform crop classification in the U.S. Corn Belt to classify corn and soybean crops with relative success. They were able to achieve an overall classification accuracy in Iowa of 82% and 75% in Illinois [8]. Hassan Bazzi et al. conducted an experiment in 2019 using random forest (RF) and decision tree (DT) on Sentinel-1 data to classify paddy rice in Camargue, France getting high overall accuracy of 96.3% with DT and 96.6% with RF [9]. Even more recently in 2022, Soma et al. used random forest (RF), gradient boosting (GB), support vector machines (SVM), and classification and regression trees (CART) in the study area of Odisha using Sentinel-1. Their results showed an overall accuracy of 98.47% RF, 98.77% CART, 98.83% Gradient Boosting and 98.26% SVM. [10].

Deep learning has shown the ability to learn robust feature representation in a wide range of fields and so the potential for utilization in multispectral data is highly viewed. However, There have been questions on the efficiency of deep learning methods. In 2020, Koppaka et al. used SVM, RF, CNN, recurrent neural network (RNN), long-short-term-memory (LSTM), and gated recurrent unit (GRU) on Sentinel-2 remote sensing image data for crop classification. In their experiment SVM had the highest accuracy of 95.9% followed by RF 91.9%, 1-D-CNN 89.32%, RNN-LSTM 89.32%, RNN-GRU 86.93%, and 2-D-CNN 83.96% [11].

In June 2017, the Transformer architecture was created by Google for the purpose of natural language processing (NLP) tasks but has since seen fast adoption in other fields like Computer Vision [12]. Transformer uses a sequence-to-sequence architecture and consists of encoder and decoder layers. The main aspect of Transformer is its usage of multi-headed self-attention. Multi-headed self-attention allows the Transformer to parse sequences and retain dependency between inputs which, according to the creators, allows the architecture to outperform recurrence and convolutions [3].

In the original study, the combination of CNN and Transformer showed a performance improvement compared with other traditional machine learning methods. The study area, Sacramento Valley in California, was used for crop classification on 10 different crop types. CNN-Transformer hybrid performed the best with an overall accuracy of 98.97%, followed by CNN-LSTM 96.69%, Multi-Temporal CNN 97.29%, SVM-RBF 94.11%, and RF-200 92.85%. Thus, CNN-Transformer improved over traditional machine learning methods RF-200 and SVM-RBF by 6.12% and 4.86% respectively [1].

2 Background

For the purpose of replicating the CNN-Transformer architecture and acquiring similar results to the original study I try to follow as closely as possible the data and parameters used. I will first introduce the platforms and data sets briefly and then describe how the data is preprocessed for usage later in the model. Then I will describe how the CNN-Transformer architecture is implemented.

2.1 Earth Engine

Google Earth Engine is a cloud platform that contains multi-petabyte analysis ready data which is able to be accessed through an API or the web-based interactive development environment (IDE) using javascript [13]. Google Earth Engine was selected over traditional remote sensing processing software such as ENVI and ArchGIS due to the convenience it offers; being able to quickly access imagery and subsequently automate a preprocessing workflow.

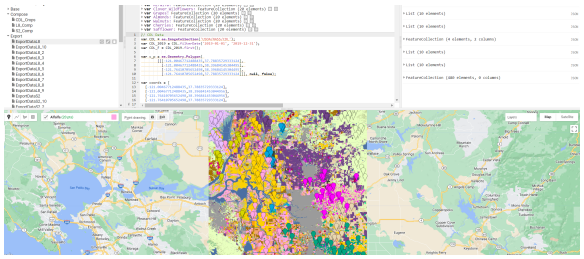

Figure 1: Earth Engine Web-based IDE

Earth Engine uses 2-D gridded raster bands to display data in an image which can then be grouped together using an image collection. This collection allows for filtering, sorting, and altering of the image data [13]. Earth Engine is used to acquire Cropland Data Layer (CDL), Sentinel-2, and Landsat-8 remote-sensing data. The remote-sensing data is then preprocessed in the web-based IDE and exported to Google Drive using GeoJSON to later use in Google Colaboratory for crop classification.

2.2 Google Colaboratory

Google colaboratory is a machine learning tool that is a web IDE released in 2017. It is a hosted Jupyter notebook on top of Python that requires no setup. Google colaboratory contains pre-installed libraries, saves to the cloud, and offers free GPU and TPU use.

2.3 Tensorflow Keras

Keras is built on top of Tensorflow which enables users to build custom deep learning models. Released in 2015, It is an open-source library that is used to create the CNN-Transformer, CNN-LSTM, Multitemporal-CNN, and CNN model that we will later use in our experiment.

2.4 Study Area

The study area selected for the dataset is Sacramento Valley, California. The bottom left corner of the region selected has Longitude/Latitude -121.764/37.788 and the area is approximately 100 km x 100 km. In this region there are large crop plantations that contain various crop types. Following the original study 10 crop types are selected in the region which are corn, rice, alfalfa, clover/wildflowers, grapes, almonds, grass/pasture, cherries, and safflower.

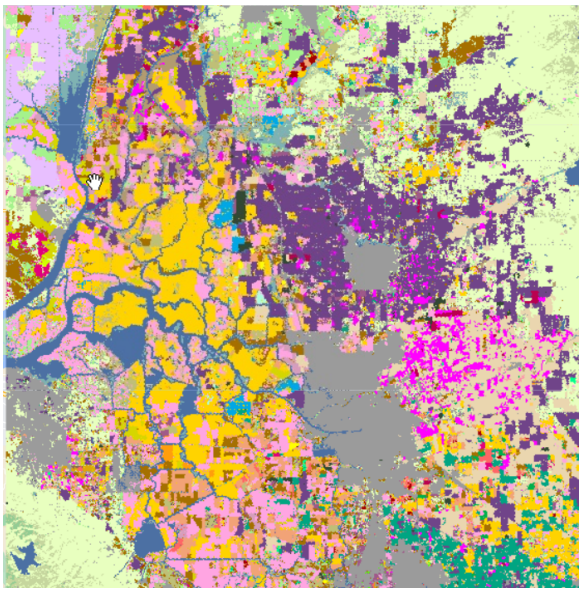

Figure 2: Cropland Data Layer of Sacramento Valley, California

2.5 Cropland Data Layer

The purpose of the CDL is to assess agricultural land across the United States and provides a raster crop-specific 30m resolution land cover data layer produced annually by the United States Department of Agriculture (USDA) National Agricultural Statistics Service (NASS) starting from 2008. CDL is produced utilizing medium resolution satellite imagery and is processed by NASS throughout the year. The final CDL released for the year represents the geospatial distribution of crops during the summer. It was shown that CDL, beginning from 2008, correctly identified cropland 97% of the time or greater on a nationwide coverage [14]. Cropland data layer (CDL) is used as the ground truth in the experiment. In Earth Engine The name of the dataset selected is USDA NASS Cropland Data Layers.

2.6 Landsat-8

Landsat-8 is a remote sensing satellite that carries 2 sensors, the Operational Land Imager (OLI) and

the Thermal Infrared Sensor (TIRS) which collect multi-band data. launched in 2013, Landsat-8 has a revisit interval of 16 days. 11 Bands are provided by Landsat-8; 9 of which come from OLI and 2 that come from TIRS. Regarding spatial resolution, Vis-NIR-SWIR (400 - 2500 nm) bands has a spatial resolution of 30m, the panchromatic band has a spatial resolution of 15m, and the Thermal IR band has a spatial resolution of 100m. In our experiment, only surface reflectance (SR), which is the light reflected from the surface of Earth, is utilized so band 9 Cirrus, which detects high-altitude clouds, and bands from TIRS, that detect heat, are not used. In Earth Engine the data set selected is USGS Landsat-8 Level 2, Collection 2, Tier 1 data. This dataset is derived from Landsat-8 OLI/TIRS sensor and contains the atmospherically corrected surface reflectance and land surface temperature.

2.7 Sentinel-2

Sentinel-2 is composed of twin satellites, Sentinel-2A and Sentinel-2B, which work together for a revisit interval of 5 days. Sentinel-2 carries an optical instrument that collects 13 spectral bands which have spatial resolution 10m, 20m, 60m. In our experiment, only surface reflectance is utilized so Band 10 Cirrus, which detects high-altitude clouds, is not used. The dataset selected in Earth Engine is Sentinel-2 MSI:MultiSpectral Instrument, Level-2A. Earth Engine computes atmospherically corrected surface reflectance data by running sen2cor on Sentinel-2 L1 assets.

3 Preprocessing

Preprocessing is done through Earth Engine and later on with Google Colaboratory. Data is acquired in Earth Engine, which is preprocessed and exported to Google Drive. Then Google Colaboratory imports the data from Google Drive and preprocesses it further into data that can be used as input for a classification model.

3.1 Earth Engine Preprocessing

3.1.1 Satellite Image preprocessing

Sentinel-2, Landsat-8 surface reflectance data and Cropland Data layer is collected from Earth Engine are preprocessed in several steps. First, Satellite imagery is preprocessed for cloud-free images. The data set is filtered on less than 30% cloud cover on remote-sensing images in the year of 2019. To remove clouds and cloud shadows in Landsat-8 we utilize the QA_PIXEL, which contain atmosphere conditions, and QA_RADSAT, which contain radiometric saturation data, from Earth Engine. With these attributes we are able to develop masks for unwanted pixels: clouds, and cloud shadows. The image bands are then scaled to the correct unit of measurement for surface reflectance. For Sentinel-2, the QA_60 band, which contains the cloud mask, is used to mask clouds and cloud shadows. Sentinel-2 image data is also scaled to get the correct unit of measurement for surface reflectance.

Starting from January to December in the year of 2019, in order to obtain a cloud-free composite image of the study region that does not contain missing pixels, composites are created every 2 months in which the median pixel of all images is used. Then the cloud-free composite image is clipped to the study region.

3.1.2 Crop Data Collection

For the collection of data in our experiment we create 10 geometry imports representing the 10 crop types we aim to classify which consist of corn, rice, alfalfa, clover/wildflowers, grapes, almonds, grass/pasture, cherries, and safflower. By inspecting the cropland data layer in the web-based IDE a cropland identifier can be acquired which we can use to locate areas where pure single crop regions exist. After identifying single crop regions, 200 points are selected for each of the 10 crops which are evenly distributed across the study region.

For each of the crop data points a sample patch is created. A sample patch consists of an area 60 x 60 m where spatial-spectral data for Sentinel-2 and Landsat-8 is collected. Depending on the spatial resolution of the band the 60 x 60 m is split into pixels which contain the respective spectral band data as described below:
Sentinel-2:

- 4 bands at 10 m: 490 nm (B2), 560 nm (B3), 665 nm (B4), 842 nm (B8).
- 6 bands at 20 m: 705 nm (B5), 740 nm (B6), 783 nm (B7), 865 nm (B8a), 1610 nm (B11), 2190 nm (B12).
- 2 bands at 60 m: 443 nm (B1), 945 nm (B9).

Landsat-8:

- 7 bands at 30 m: 435 nm (B1), 452 nm (B2), 533 nm (B3), 636 nm (B4), 851 nm (B5), 1566 nm (B6), 2107 nm (B7)

There are 4 spatial resolutions which can be split accordingly: 6x6 pixels for 10m bands, 3x3 pixels for 20m bands, 4x4 pixels for 30 m bands, and 1x1

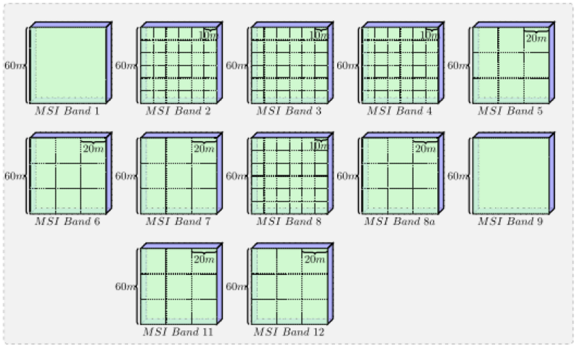

Figure 3: Sentinel-2 Multiband MSI images [1]

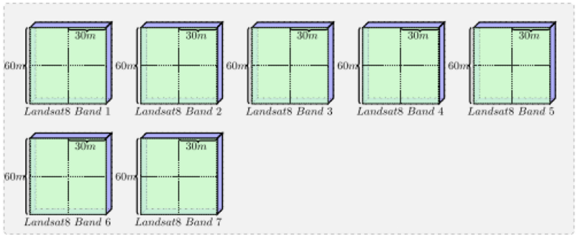

Figure 4: Landsat-8 Multiband OLI images [1]

pixel for 60 m bands. In order to create a sample patch for each data point spatial-spectral information needs to be collected for every pixel in the 60 x 60 m area. First the data point is designated as the center of the sample patch and then we store the cardinal direction coordinates for each pixel in the patch. With the cardinal direction coordinates we can get the area of the pixel to collect the spectral data for bands of the same spatial resolution. This can be done through zonal statistics in Earth Engine in which we specify the spectral data be averaged over the pixel area. Once sample patches are collected on all points in the data set we then export it to Google Drive using a GeoJSON format. For each crop and spatial resolution data is exported separately.

3.2 Colaboratory Preprocessing

Preprocessing is done in colaboratory to shape input data that can be used for the CNN-Transformer model. Data is imported from Google Drive that we use to store the crops, spatial resolutions, 2-month intervals, and spatial-spectral information altogether. Landsat-8 and Sentinel-2 data are then combined into one data set. Spatial-spectral information is taken as a sequence and so we reshape it into a band image according to the pixels in the sample patch into a 3-D array (w, h, c). where w and h correspond to the width and height of the pixels which were split according to the respective spatial resolution and c refer to the bands of the same spatial resolution. Each of the 2 month intervals and spatial resolutions are separated to be used as input for a total of 24 inputs for the CNN-Transformer model.

4 CNN-Transformer Architecture

The CNN-Transformer architecture consists of several modules: the spatial-spectral scale unification features extraction module, position feature module, multilayer transformer encoder module, feed forward module, and the softmax output layer module. Tensorflow Keras is used to build the CNN-Transformer architecture. A sample in the study region can be expressed as $M = [m_1, m_2, m_i, ..., m_6]$, where i corresponds to a specific 2 month interval in the year.

For sample M, m_i contains the spatial-spectral data for one of the four spatial resolutions in Sentinel-2 and Landsat-8. This can be expressed as $m_i = [b_{i_1}, b_{i_k}, ..., b_{i_4}]$ where k represents the different spatial resolutions (10m, 20m, 30m, 60m) and b_{i_k} refers to the band image which is a 3-D array (w, h, c) that contains the spatial-spectral band data as mentioned in the preprocessing section.

4.1 Multisensor Spatial-Spectral Scale Unification

Spatial-Spectral Unification (SSU) consists of two steps. First spatial unification is done where transposed convolution is applied on each band image as a way to normalize the width and height to 6x6 which matches the largest spatial resolution 60m. Then spectral unification uses convolution on the spectral bands to normalize the number of bands. Since Sentinel-2 and Landsat-8 data is already combined spectral unification is not a necessary step however keeping in line with the original study, the spectral unification step is kept in addition to batch normalization after every convolution.

4.1.1 Spatial Unification

To perform spatial unification, transposed convolution is applied to all the band images in order to normalize the image dimension (width and height) to a 6x6. Once spatial unification is performed the band images are then concatenated together:

$$\begin{aligned} B_{i_k} &= ConvT(b_{i_k})b_{i_k} \\ S_i &= Concat(B_{i_k} : k \in 1, ..., 4) \end{aligned} \tag{1}$$

In the formula ConvT represents transposed convolution on b_{i_k} to perform spatial unification. Concat corresponds to concatenation of the band images along the spatial resolutions as the image dimensions are now normalized. S_i is a 3-D tensor of dimensions (w_i, h_i, c_i), where i indicates the unified spatial scale. There must be different parameters for each spatial resolution to apply transposed convolution as described below:

- 10 m resolution, Transposed Convolution using 1 × 1 kernel over a 6 × 6 input with unitary stride and no padding, output is a 6 × 6.
- 20 m resolution, Transposed Convolution using 4 × 4 kernel over a 3 × 3 input with unitary stride and no padding, output is a 6 × 6.
- 30 m resolution, Transposed Convolution using 5 × 5 kernel over a 2 × 2 input with unitary stride and no padding, output is a 6 × 6
- 60 m resolution, Transposed Convolution using a 6 × 6 kernel over a 1 × 1 input with unitary stride and no padding, output is a 6 × 6.

4.1.2 Spectral Unification

After spatial unification is done spectral unification normalizes the number of bands.

$$E_i = Flatten(Conv(S_i)) \tag{2}$$

Conv indicates a 2-D convolution using 1 x 1 kernel with unitary stride and no padding with 5 filters, output is a 6 x 6 x 5. This output is the 3-D SSU tensor which we then apply Flatten to turn the 3-D tensor into a 1-D tensor. E_i represents the 1-D SSU tensor for the specific 2 month interval.

4.2 Position Feature Embedding

Position Feature Embeddings allows us to encode a temporal aspect to the SSU data. Position Feature Embedding formula is described as below [1]:

$$\begin{aligned} PE_(p, 2i) &= sin(p/10000^{2i/d_model}) \\ PE_(p, 2i+1) &= cos(p/10000^{2i/d_model}) \end{aligned} \tag{3}$$

Where p indicate the sequence which has length of 6 corresponding to the 2 month intervals and i is the dimension of the position feature. d_{model} is length of the 1-D SSU tensor which is 6x6x5=180. After positional embedding is acquired then we add this to the SSU tensors. We first concatenate all SSU tensors according the the 2 month interval and then directly add the Positional Feature Embedding as below:

$$\begin{aligned} W_E &= Concat(E_i; i \in 1, 2, ..., 6) \\ h_0 &= W_E + PE \end{aligned} \tag{4}$$

W_E is the SSU feature sequence and PE is the position feature embedding. h_0 is the result of the addition of the SSU feature sequence and position feature embedding.

4.3 Multilayer Transformer Encoder Module

CNN-Transformer consist of 4 stacked encoder blocks. Each encoder block utilizes multi-headed self-attention followed by a fully connected feed-forward layer network. Layer normalization and residual connection is applied in each block. Transformer is a sequence-to-sequence architecture and so h_0 is reshaped into $[6, 180]$ to obtain 6 sequences that correspond to the 2 month intervals and SSU unified data with the Position Feature Embedding. The multilayer encoder module can be described as below:

$$h_l = tranformer\_encoder(h_0) \tag{5}$$

where h_l is the output of the transformer. The transformer encoder uses 6 heads, a head size of 30, and a feed-forward dimension of 180.

4.4 Feed-Forward and Softmax Output Layer Module

We pass the transformer encoder output to the feed-forward layer to obtain a prediction which is a crop label from the softmax output layer. First we flatten h_l and then pass it to the feed-forward layer which consists of a dense layer of 100 nodes connected to a dense layer of 40 nodes. Then this feed-forward layer is passed to the softmax output layer which predicts a crop label which is one of

the 10 crop types used in the data set. This process can be described as follows:

$$P(y|G_1, ..., G_6) = softmax(h_l W_y) \quad (6)$$

In which we maximize a standard language modeling objective [1]

$$L(C) = \sum_{(W_E, y)} \log P(y|G_1, ..., G_6) \quad (7)$$

Where W_y is the feed-foward layer parameters and y is the ground truth of the sample. Using the conditional probability P we maximize a standard language modeling objective $L(C)$.

5 Experiment

The study region where the data set was collected is Sacramento Valley, California in the year of 2019. The data set is composed of spatial-spectral multiband data of both Sentinel-2 and Landsat-8 which were preprocessed into 2000 samples. Each of the 10 crop types contain 200 samples. The 10 crop types are: corn, rice, alfalfa, clover/wildflowers, grapes, almonds, grass/pasture, cherries, and safflower.

The goal of our experiment is to have the classification model to correctly predict the ground truth obtained from cropland data layer. I conducted the experiment with 2 train test splits. The first train test split I used was 25% train evenly split across the crop types and 75% test on the remaining data. The second train test split I used was 37.5% train evenly split across the crop types and the remaining 62.5% data was used for testing. The reason two splits were used was to observe if there was any difference in performance if models had more training data.

5.1 Evaluation Criteria

In order to measure the performance of the classification models overall accuracy and Kappa coefficient was used as the evaluation criteria. Overall accuracy (OA) and Kappa Coefficient are derived from the confusion matrix which shows the performance measure of the classification model:

- Overall Accuracy (OA): Overall accuracy takes the sum of correctly predicted values over the total number of samples used for testing. This measures the percentage of correctly predicted values.
- Kappa Coefficient: Calculated with scikit-learn library using cohen kappa score. It measures the inter-annotator agreement. It is described as the consistency measure between ground-truth map and final classification map.

5.2 Classification models

In addition to the CNN-Transformer, I implement the same classification models used in the original study. An ablation study is also performed by removing the Multilayer Transformer Encoder module, utilizing just the CNN and Feed Forward modules for crop classification which I refer to as the CNN crop classifier. The models used in the experiment are described as below:

- RF-200: Number of decision trees in random forest set to 200.
- SVM-RBF: Support vector machine classification with the RBF kernel. Fivefold cross-validation is also used to determine the best model.
- CNN-Transformer crop classifier: The implemented CNN-Transformer architecture
- CNN crop classifier: From the implemented CNN-Transformer model, this simply removes the Multilayer Transformer Encoder module.
- Multitemporal CNN crop Classifier: From the implemented CNN-Transformer Model, this removes the Multilayer Transformer Encoder module and replaces it with CNN layers.
- CNN-LSTM crop classifier: From the implemented CNN-Transformer Model, this removes the Multilayer Transformer Encoder module and replaces it with LSTM layers.

For implementing RF-200 and SVM-RBF the machine learning library scikit-learn was used. Since SSU is not performed on the traditional machine learning methods I instead flatten the spatial-spectral data for a sample patch into one sequence to serve as input.

The deep learning models are optimized with the Adadelta algorithm using a learning rate of 0.01 for training. Training Epochs is set to 1000 and the number of CNN and LSTM layers in the Multitemporal CNN and CNN-LSTM is set to 4 the same amount of encoders that are stacked in the Multilayer Transformer Encoder module [15].

Class No.	Class	RF-200	SVM-RBF	CNN-Transformer	CNN	Multi-Temporal CNN	CNN-LSTM
0	Corn	96.00	87.33	64.67	67.33	70.00	55.33
1	Grassland/Pasture	93.33	99.33	93.33	91.33	91.33	90.00
2	Rice	96.00	70.00	68.00	63.33	62.87	56.00
3	Alfalfa	88.67	88.00	86.00	82.67	78.00	70.00
4	Clover/Wildflowers	92.00	90.00	74.00	60.67	70.67	36.00
5	Grapes	92.67	91.33	82.00	83.33	82.00	74.67
6	Almonds	89.33	80.00	68.00	64.00	67.33	64.67
7	Walnuts	99.33	99.33	91.33	93.33	89.33	88.00
8	Cherries	87.33	74.67	80.00	71.33	72.67	59.33
9	Safflower	93.33	86.67	76.67	75.33	71.33	56.67
OA	-	92.80	86.67	78.40	75.27	75.33	65.07
KAPPA	-	0.92	0.8519	0.76	0.7252	0.7281	0.6119
Time		3.36s	1.55s	23min 15s	14m 27s	14m 14s	40m 17s

Table 1: Classification Accuracies of Different Models for 25% train / 75% test split

Class No.	Class	RF-200	SVM-RBF	CNN-Transformer	CNN	Multi-Temporal CNN	CNN-LSTM
0	Corn	95.20	90.40	76.80	81.60	72.00	69.60
1	Grassland/Pasture	95.20	97.60	94.40	93.60	91.20	88.00
2	Rice	97.60	72.00	76.00	71.20	74.40	69.60
3	Alfalfa	91.20	87.20	90.40	92.00	84.80	79.20
4	Clover/Wildflowers	96.00	92.00	79.20	78.40	74.40	61.60
5	Grapes	94.40	92.00	92.00	92.80	90.40	77.60
6	Almonds	91.20	90.40	63.20	64.00	65.60	56.80
7	Walnuts	98.40	99.20	90.40	89.60	84.80	85.60
8	Cherries	88.80	76.00	82.40	80.80	73.60	52.80
9	Safflower	92.00	92.80	81.60	79.20	75.20	71.20
OA	-	94.00	88.96	82.64	82.32	78.64	71.20
KAPPA	-	0.9333	0.8519	0.8071	0.8036	0.7627	0.6799
Time		3.53s	3.54s	30min 4s	22m 49s	22m 10s	1h 16s

Table 2: Classification Accuracies of Different Models for 37.5% train / 62.5% test split

6 Results

Two splits were conducted in this experiment. I first use a 25% train and 75% test split and then a 37.5% train and 62.5% test split. The classification confusion matrix and crop classification table for the train test splits are shown in Fig. 5, Fig. 6 and Table 1, Table 2 respectively.

6.1 train 25% and test 75% split

The confusion matrix and the crop classification table for this split are shown in Fig. 5 and Table 1. From the total of 2000 samples in the dataset, 500 are chosen as training samples with the remaining 1500 as test samples. Training samples are split evenly across the crop types, thus 50 samples for each crop type are used to train the model. Observing the crop classification table the traditional machine learning models outperform the deep learning models. The best performing model is RF-200 with an overall accuracy of 92.80% and Kappa Coefficient of 0.92. The second best performing model is then SVM-RBF which has an overall accuracy of 86.67% and a Kappa Coefficient of 0.8519. CNN-Transformer compared to RF-200 has a lower overall accuracy of 14.40%, and lower Kappa coefficient by 0.16. When compared to the other deep

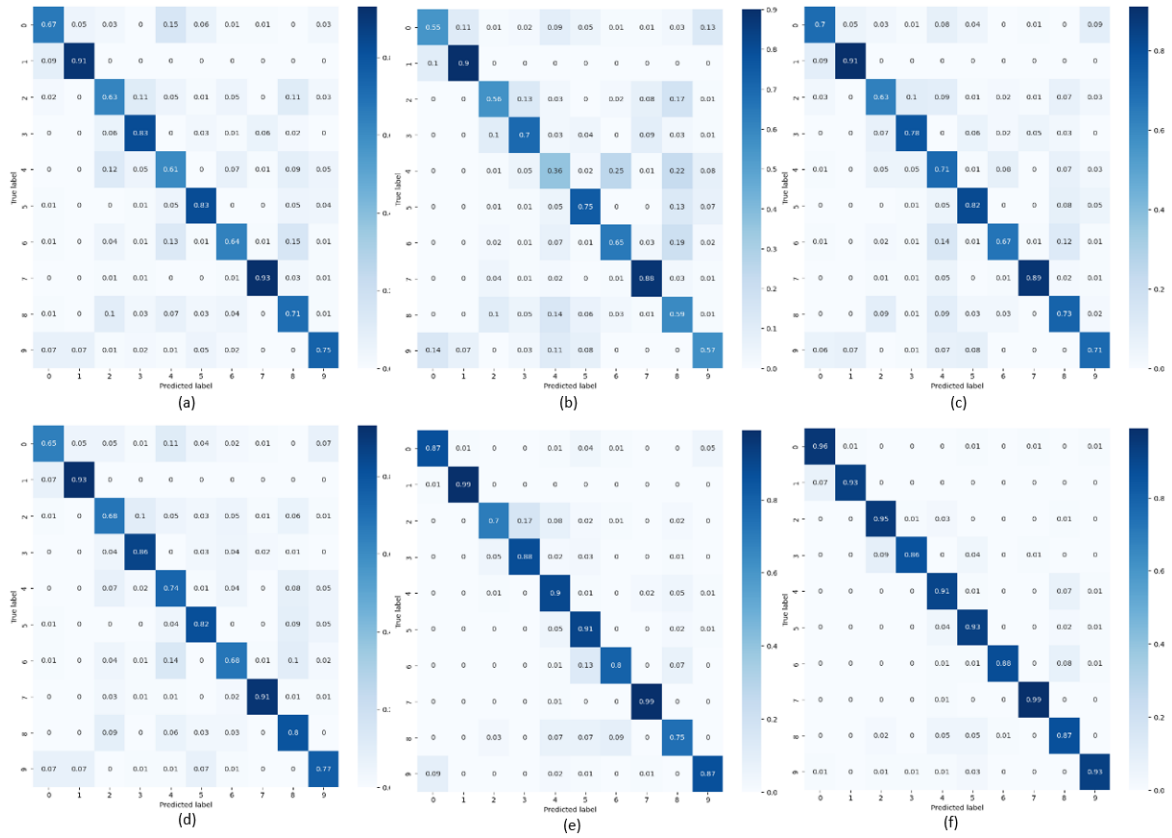

Figure 5: Confusion matrix of different models with train 25% and test 75%. (a) CNN (b) CNN-LSTM (c) Multitemporal CNN (d) CNN-Transformer classifier (e) SVM-RBF (f) RF-200

learning models CNN-Transformer does perform the best. CNN-Transformer has an overall accuracy of 78.40% and Kappa coefficient of 0.76. CNN and Multi-Temporal CNN performed similarly with an overall accuracy of 75.27%, 75.33% and Kappa Coefficient 0.7252, 0.7281 respectively. CNN-LSTM performed the worst over all models with an overall accuracy of 65.07% and Kappa coefficient of 0.6119.

Looking at the confusion matrix the CNN-Transformer model's lowest classification accuracy is on corn at 65% which misclassifies corn most with clover/wildflowers and safflowers. Looking at the best performing model RF-200 the lowest classification accuracy is Cherries at 87% which compared to the other models is on the higher end of classification accuracy as taking a look at the CNN-LSTM model the lowest classification accuracy is 36% with Clover/Wildflowers. CNN and Multi-Temporal CNN models share similar classification accuracies.

6.2 train 37.5% and test 62.5% split

The confusion matrix and the crop classification table for this split are shown in Fig. 6 and Table 2. From the total of 2000 samples in the dataset, 750 are chosen as training samples with the remaining 1250 as test samples. Training samples are split evenly across the crop types, thus 75 samples for each crop type are used to train the model. Observing the crop classification table, RF-200 is still the best performing model increasing overall accuracy marginally by 1.2%, and Kappa Coefficient by 0.1333 from the previous split. The only change in the models is CNN outperforming Multi-Temporal CNN in overall accuracy by 3.68% and Kappa Coefficient by 0.0409. CNN-Transformer does show a level of improvement by increasing overall accuracy by 5.97%, and Kappa Coefficient by 0.0471. However this is also matched by CNN which increases overall accuracy by 7.05% and Kappa Coefficient by 0.784. From the results the difference in performance between the CNN-Transformer model and the CNN model is slim with CNN-Transformer having a slight improvement of 0.32% in overall accuracy and 0.0035 in Kappa Coefficient. CNN-LSTM model performed worst over all models again with an overall accuracy of 71.29% and Kappa coefficient of 0.6799.

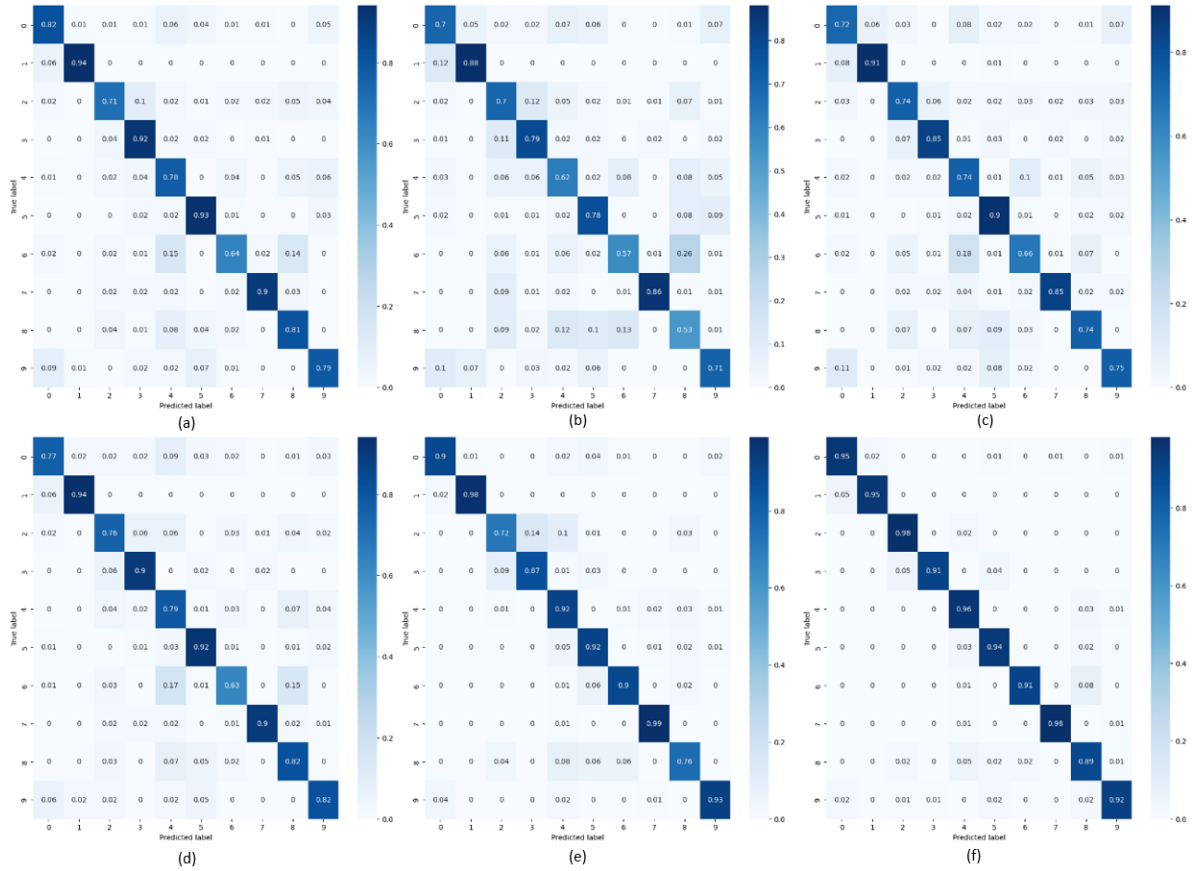

Figure 6: Confusion matrix of different models with train 37.5% and test 62.5% (a) CNN (b) CNN-LSTM (c) Multitemporal CNN (d) CNN-Transformer classifier (e) SVM-RBF (f) RF-200

Looking at the confusion matrix the CNN-Transformer model improved corn classification rate by 12% from the previous split, however, now the lowest classification accuracy is 63% with Almonds which which is a 5% decrease in classification accuracy. For the RF-200 model the worst performing crop is alfalfa with a classification accuracy of 86% and cherries had a slight improvement in classification accuracy now at 87%. The CNN and CNN-LSTM models overall improved classification accuracies across the crops while in Multitemporal CNN model there was only slight increases to the classification accuracy for some of the crops.

6.3 Training Time

In the experiment models were trained in Google Colaboratory using free computing resources. Training times for models were measured and shown in Table 1 and Table 2. From the times we find that the deep learning models took significantly longer to train than traditional machine learning models.

7 Differences in Approach

In this section I will address the biggest discrepancies between my approach and the original study. The differences start from the acquisition of the data. In the original study data is collected from Sentinel-2 Level 1C data from USGS which is then processed through Sen2cor atmospheric correction to convert into surface reflectance. For Landsat-8 Level1 DN data is collected which is converted to surface reflectance via atmospheric tool FLAASH.

In Earth Engine, Landsat-8 SR data set was created with Land Surface Reflectance Code (LaSRC) and Sentinel-2 SR data set was created by running sen2cor. Earth Engine does mention that not all L2 data was produced from L1 so missing data is a concern. In the original study they use the year 2018 to collect samples from the study region. Using Earth Engine surface reflectance data for Sentinel-2 from the date range of June-September for 2018 is missing which leads me to instead use the year of 2019 for my experiment.

Another difference is with the bands used for the satellite imagery. In the original study, for Landsat-

8 they claim there are 11 bands, of which only Band 9 Cirrus is excluded, and that 9 multispectral bands at 30 m and 1 panchromatic band at 15 m exist. This is incorrect as 2 of the 11 bands are obtained from TIRS which has a spatial resolution of 100 m. I exclude the TIRS bands and the panchromatic band, which is missing in Earth Engine, so I only use bands 1-7 from Landsat-8.

A departure in approach from the original study could be found in how temporal profiles were created. In the original study, self-organizing Kohonen maps (SOM) is used to fill missing pixels in time series data from Sentinel-2 and Landsat-8. A total of 65 Temporal profiles were selected over 2018 where a complete cloud-free mosaic from either Sentinel-2A, Sentinel-2B, or Landsat-8 could be created.

In my study, SOM was not used and instead a pixel median for every 2 month interval was taken to create a cloud-free image composite. This resulted in 12 unique temporal profiles; 6 coming from Sentinel-2 and the other 6 from Landsat-8. Since the intervals are the same for both Sentinel-2 and Landsat-8 the temporal profiles were combined so only 6 temporal profiles remained. The reason 2 month intervals is used is because of the revisit interval of Landsat-8 being 16 days. If 1 month intervals were chosen the image composite would be incomplete. Possibly a lower interval for Sentinel-2 could be chosen as the revisit interval is 5 days since it combines both Sentinel-2A and Sentinel-2B, however, I made the decision to keep the interval the same as Landsat-8 so the temporal profiles could be combined.

In the original study it was mentioned that there are differences in crop phenology and that sparse RS data are not able to indicate subtle differences [1]. Thus, it is possible that I am not able to take advantage of these phenological differences since only 2 month intervals were utilized which could have lowered the overall performance of my deep learning models.

Another difference is with the samples in the dataset. In the original study they were able to acquire 39959 total samples. The selection process of these samples was not extensively mentioned. 1% of the data is used for training and the remaining 99% for verification. So only 399 values are used to train the classification models with crops like Rice only using 22 training samples. Even with the low amount of training data CNN-Transformer was able to achieve an impressive overall accuracy of 98.97% which performed the best over all other classification models in this study.

In my study I acquired 2000 total samples which I manually selected. Samples were selected across the study region which might explain why the models performed poorly against the original study. I was unable to obtain a high classification accuracy with CNN-Transformer only getting a 78.40% overall accuracy on 25%/75% train test split and a 82.64% on 37.5%/62.5% train test split.

8 Discussion and Conclusion

In this article I implemented a CNN-Transformer hybrid architecture and compared the model on crop classification using a multitemporal multisensor dataset in Sacramento Valley, California in the year 2019. 12 temporal profiles are obtained by 2 month intervals in which the median pixel is taken to create composite cloud-free images for Sentinel-2 and Landsat-8. Since Sentinel-2 and Landsat-8 uses the same intervals the respective temporal profiles are combined. Spatial-Spectral unification is then performed on the temporal profile and uses transposed convolution to normalize the image resolution. Then spectral unification uses convolution to normalize the number of bands. After SSU data is acquired, positional encoding is then added and the output is fed to a multilayer encoder transformer that is composed of 4 encoder blocks. Classification is then done by passing the output from transformer to a feed forward layer network which is connected to a softmax output layer where the crop type is then predicted.

From my results I found that traditional machine learning models RF-200, and SVM-RBF had better performance over the deep learning models: CNN-Transformer, CNN, CNN-LSTM, and CNN-Multitemporal. RF-200 performed the best over the two train test splits that I conducted and CNN-LSTM performed the worst over all the other models. CNN-Transformer did perform better than other deep learning models in overall accuracy and Kappa coefficient, however, on the second split compared to the CNN model there was only a 0.32% difference in the overall accuracy. This seems to suggest that the transformer encoder module had little effect on improving classification accuracy in that train test split.

I was unable to replicate the results shown in the original study. In the original study they found

that CNN-Transformer performed the best over all models and RF-200 actually performed the worst. In my case RF-200 was the best performing model and CNN-Transformer fell behind both RF-200 and SVM-RBF for both train test splits. This difference in results could possibly be due to a lot of factors which one of them being the implementation of the CNN-Transformer architecture.

If I were to implement any changes to my implementation of the CNN-Transformer architecture I would try to include more multitemporal profiles taking into account the shorter revisit interval of Sentinel-2 and also take a look at using SOM to fill in missing pixel data. One other aspect I would change would be in the Transformer module. The Multilayer Transformer Encoder module only uses 4 stacked encoder layers, perhaps that could be adjusted to improve classification accuracy. By implementing the changes it would be beneficial to see how close the results can compare to the original study and if the CNN-Transformer model can overcome traditional machine learning models. Ultimately, in my case I was unable to see if the transformer architecture truly improved performance in crop classification.

References

[1] Tianxu Zhang Zhengtao Li, Guokun Chen. A cnn-transformer hybrid approach for crop classification using multitemporal multisensor images. 2020.

[2] Raphael Couturier Stephane Cuenat. Convolutional neural network(cnn) vs vision transformer(vit) for digital holography. 2022.

[3] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. 2017.

[4] R. Mueller M. Craig C. Boryan, Z.W. Yang. Monitoring us agriculture: The us department of agriculture national agricultural statistics service cropland data layer program. 2011.

[5] David P. Roy Jian Li. A global analysis of sentinel-2a sentinel-2b and landsat-8 data revisit intervals and implications for terrestrial monitoring. 2017.

[6] Enrique J. Montero Herrero Joeri van Wolvelaer Manfred Keil Horst Weichelt Antonio Garzon Erik Zillmann, Adrian Gonzalez. Pan-european grassland mapping using seasonal statistics from multisensor image time series. 2014.

[7] Z. Chao Y. Wei Z. Qiang, Q. Yuan. Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network. 2018.

[8] Bakhyt Akhmedov Paul C. Doraiswamy, Alan J. Stern. Crop classification in the u.s. corn belt using modis imagery. 2007.

[9] Mohammad El Hajj Mehrez Zribi Dinh Ho Tong Minh Emile Ndikumana Dominique Courault Hatem Belhouchett Hassan Bazzi, Nicolas Baghdadi. Mapping paddy rice using sentinel-1 sar time series in camargue, france. 2019.

[10] Dayal Kumar Behera Shreela Dash Soma Gupta, Satarupa Mohanty. Machine learning based crop classification with sentinel-1 data. 2022.

[11] Teng-Sheng Moh Ravali Koppaka. Machine learning in indian crop classification of temporal multispectral satellite image. 2020.

[12] Martin Jaggi Jean-Baptiste Cordonnier, Andreas Loukas. On the relationship between self-attention and convolutional layers. 2019.

[13] Mike Dixon Simon Ilyushchenko David Thau Rebecca Moore Noel Gorelick, Matt Hancher. Google earth engine: Planetary-scale geospatial analysis for everyone. 2017.

[14] Holly K. Gibbs Tyler J. Lark, Ian H. Schelly. Accuracy, bias, and improvements in mapping crops and cropland across the united states using the usda cropland data layer. 2021.

[15] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.