Argentina Slum Classification and Identification

Xiaoran Wang | Yujing Wu | Hanyong Xu
MUSA 650 Final Project

Slum

A overcrowded, densely populated urban residential area consisting of inadequate infrastructure and insufficient living area, inhabited by low-income urban residents.

Background

- Inadequate developments of infrastructure and rising living costs accompanied rapid urbanization.
- Low-income community cannot afford formal housing.
- People who live in slums suffers from poor living conditions
  - Poor quality housing unit, insufficient living area, low accessibility to utility, poor tenure security etc.
- Currently, around **one-quarter** of the world’s population lives in slums.

Motivation

The slum-identification within a given city not only allows the government and other organizations to provide assistance and management accordingly but also serves as a metric for the effect of the overall city planning efforts.

Objective

Identify the slum areas in the city of Buenos Aires, Argentina through remote sensing images with machine-learning approaches.

Process

We will first use satellite images from two districts in Argentina: Buenos Aires and Córdoba, to train a machine-learning algorithm that identifies which regions are slums, then we will use the algorithm to detect slums of the Buenos Aires over the span of past few years using the district's satellite images of different years.

Data - Training Data

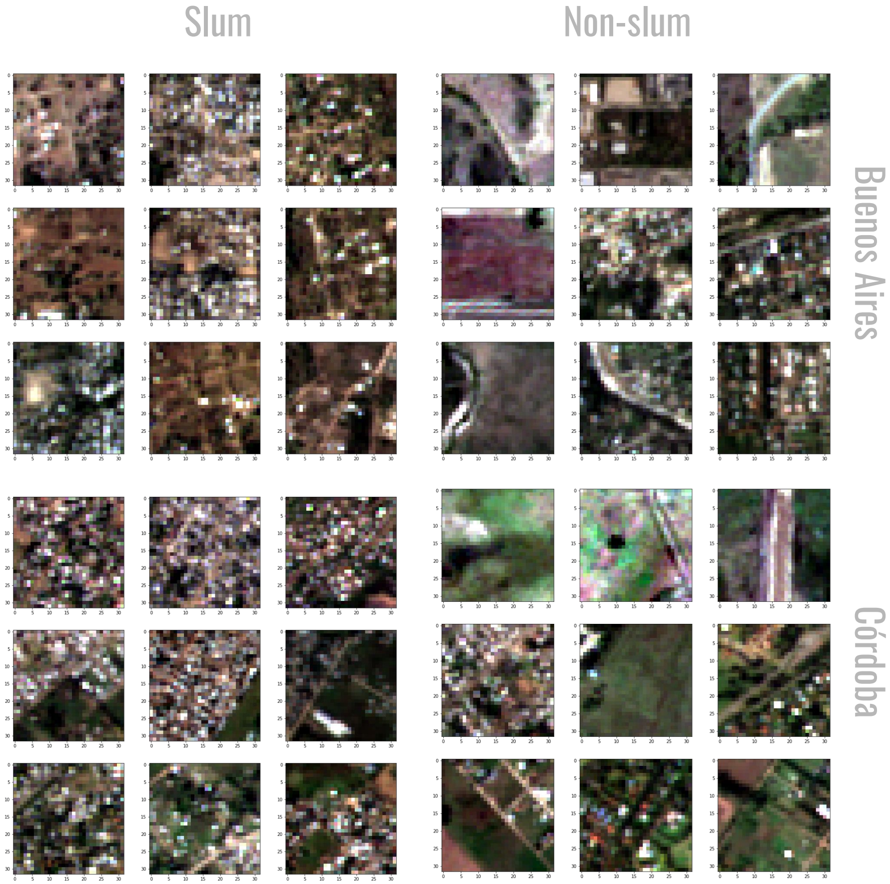
Kaggle Competition Data:

Slums and informal settlements detection

The dataset contains images of urban slums for Buenos Aires and Córdoba. The image of Cordoba was taken on 2017-06-09 (37 out of 13,004 images are labeled as slums) and the images of Buenos Aires on 2017-05-04 (1,008 out of 46,047 images are labeled as slums).

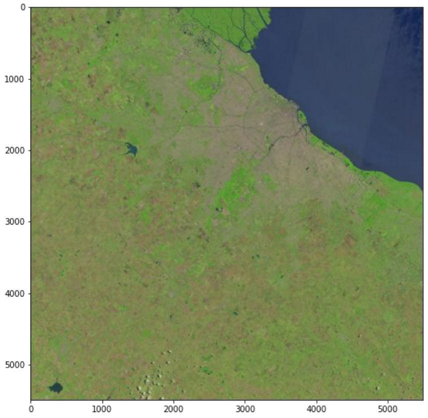
The numbers of positive and negative labeled images are heavily unbalanced.

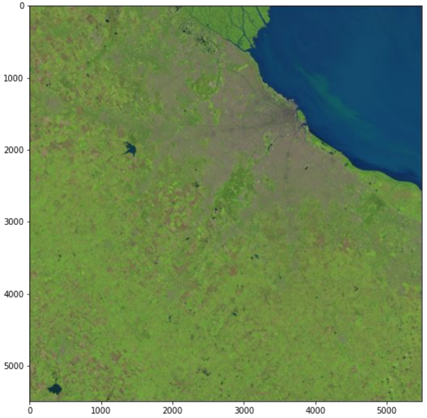
Each image comes from the Sentinel-2 sensor, with 32 by 32 pixels, 4 bands (bands 2, 3, 4, 8A), and 10-meter resolution. Images are in .tif format.



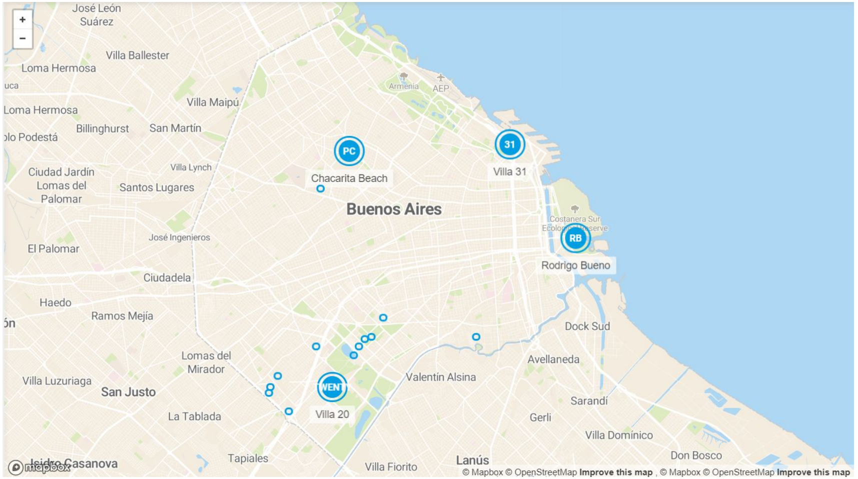
Data - Data for Application

Sentinel-2 images of the region taken on December 20th, 2017 and April 18th, 2020 from from USGS Earth Explorer





After training our model, we applied it to these two images to examine if the slums and informal settlements expanded, gradually disappeared, or stayed the same between 2017 and 2020. In addition, we also compared our findings to the maps created by Caminos de la Villa, a project that tracks the location and development of slums and informal settlements in Buenos Aires.



Support Vector Machine

Data Cleanup

- Balanced the number of images in two classes
- Scaled the data to values between 0-255 and combined 3 RGB channels into one gray image array
- Split the data into a training dataset (80% of the data) and a testing dataset (20% of the data)

Trained the model

- GridsearchCV from the following combinations:
- kernel: ‘rbf', 'poly', 'sigmoid'
- gamma: 1e-3, 'scale'
- C: 1, 2, 3, 10, 100.

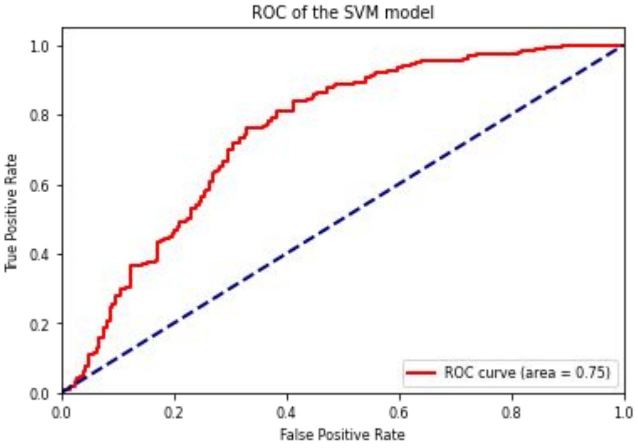
- Best parameters: ‘rbf’, ‘scale’, and C=1.

Applied the model to satellite imageries

- Cut the satellite imageries into smaller pieces of the size of our training data (32 * 32 pixels)
- Applied our model to these smaller images
- Showed the predicted slums on the larger satellite imageries

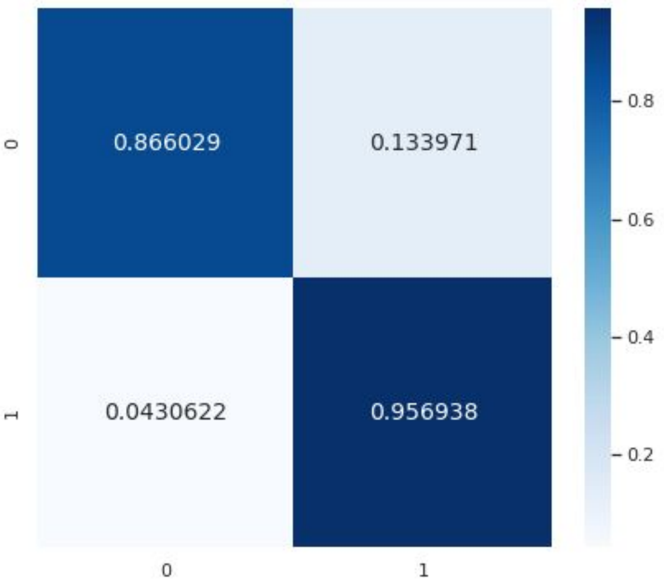
Support Vector Machine

Accuracy & AUC



The accuracy of the model is 0.68 while the AUC is 0.75.

Confusion Matrix



High TP and TN

SVM predicts urban slums well in the given Kaggle data

Support Vector Machine

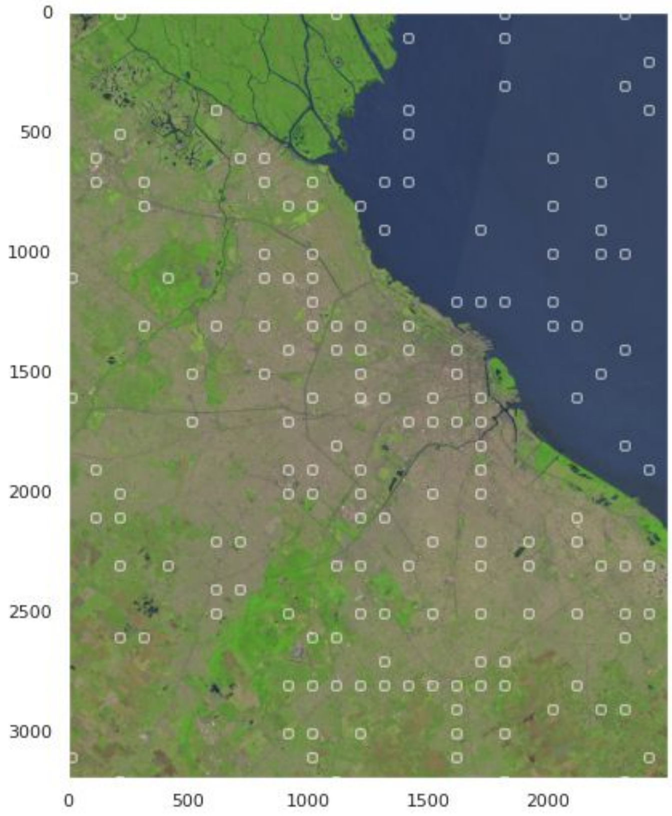


Image of Buenos Aires in 2017

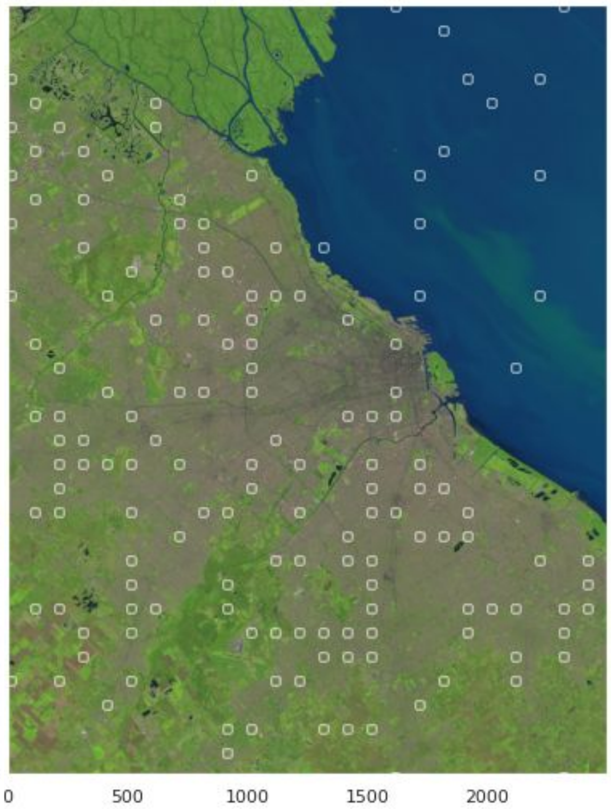


Image of Buenos Aires in 2020

These images show the areas predicted to be slums by the SVM model in and around Buenos Aires. From 2017 to 2020, the predictions indicated an decrease in the number of slums in the region.

SVM fails to separate ocean images with the land images. If we exclude those patches, the number of positive prediction increased over time.

Convolutional Neural Networks

Data Cleanup → **Trained the model** → **Applied the model to satellite imageries**

Data Cleanup

- Balanced the number of images in two classes
- Normalized the data by dividing the image data by the maximum pixel value of the training data
- Split the data into a training dataset (60% of the data) and a testing dataset (40% of the data)

Trained the model

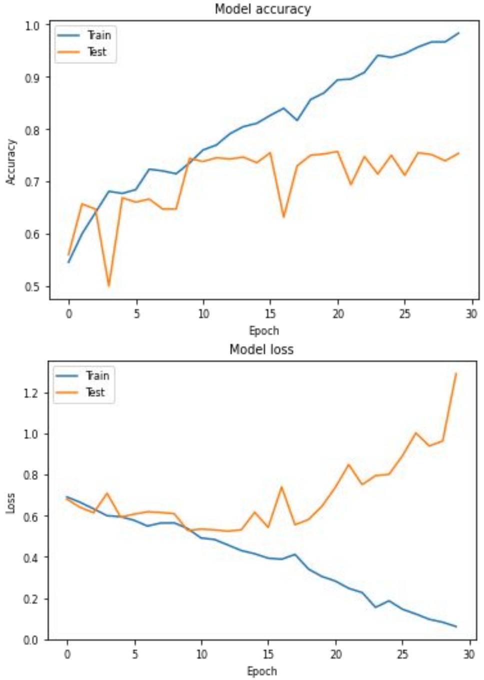
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 64)        1792
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               4194816
_________________________________________________________________
dropout_2 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1026
=================================================================
Total params: 4,308,418
Trainable params: 4,308,418
Non-trainable params: 0
_________________________________________________________________
```

Applied the model to satellite imageries

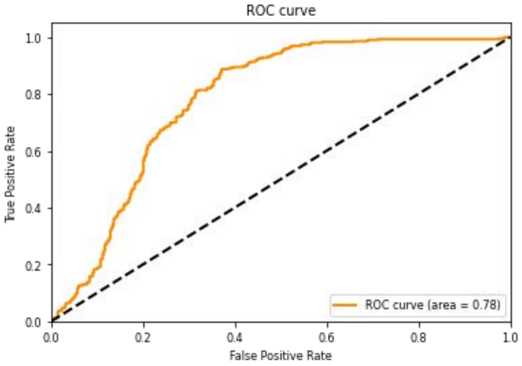
- Cut the satellite imageries into smaller pieces of the size of our training data (32 * 32 pixels)
- Applied our model to these smaller images
- Showed the predicted slums on the larger satellite imageries

Convolutional Neural Networks

Training History



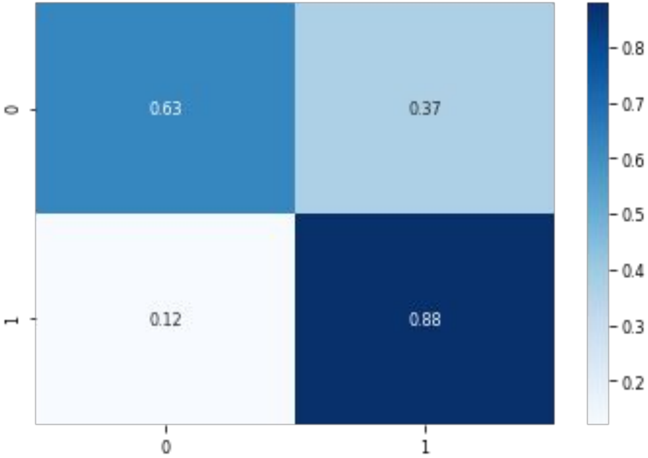
Accuracy & AUC



The accuracy of the model is 0.754 while the AUC is 0.78.

Confusion Matrix

<u>High TP:</u> CNN works well in identifying the slum images.
<u>Low TN:</u> CNN has difficulty identifying non-slum images..



Convolutional Neural Networks

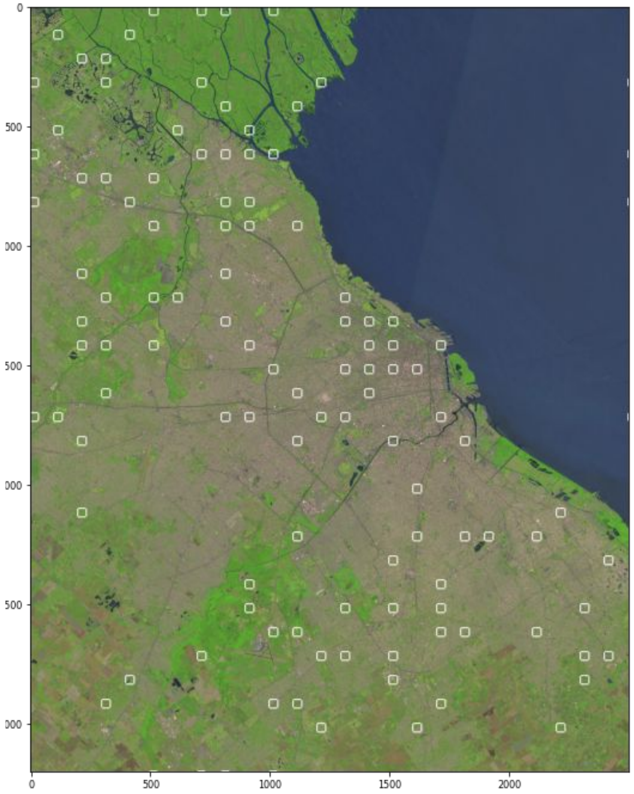


Image of Buenos Aires in 2017

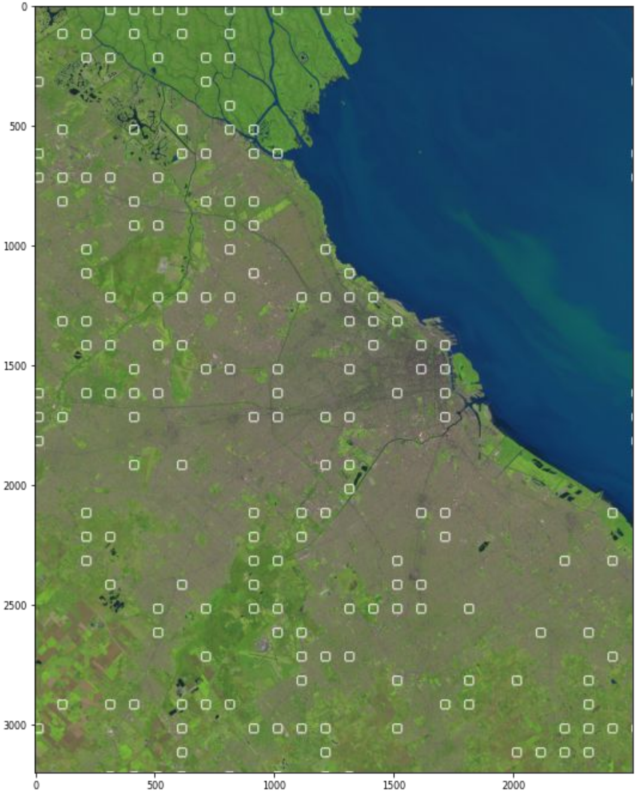


Image of Buenos Aires in 2020

These images show the areas predicted to be slums by the CNN model in and around Buenos Aires. Within the city, the model successfully identifies the slums in eastern and center Buenos Aires.

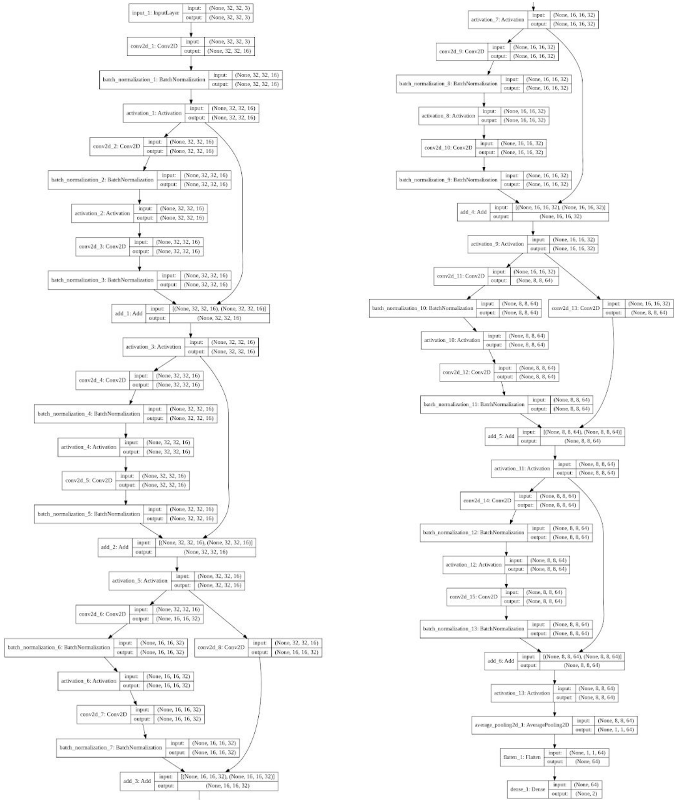
From 2017 to 2020, the predictions indicate an increase in the number of slums in the region.

Residual Networks

Data Cleanup

- Balanced the number of images in two classes
- Normalized the data by dividing the image data by the maximum pixel value of the training data
- Split the data into a training dataset (60% of the data) and a testing dataset (40% of the data)

Trained the model



Applied the model to satellite imageries

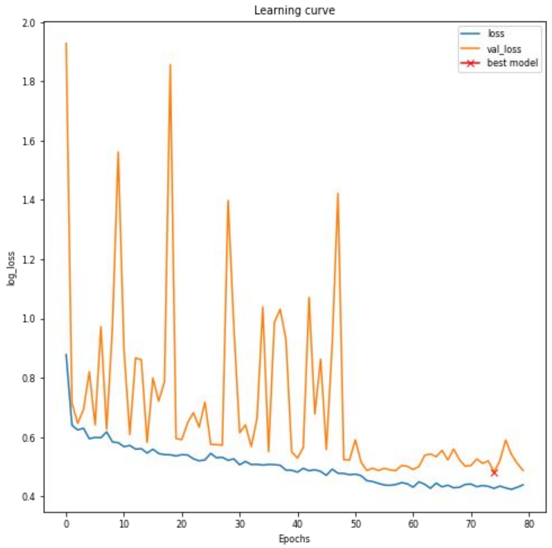
- Cut the satellite imageries into smaller pieces of the size of our training data (32 * 32 pixels)
- Applied our model to these smaller images
- Showed the predicted slums on the larger satellite imageries

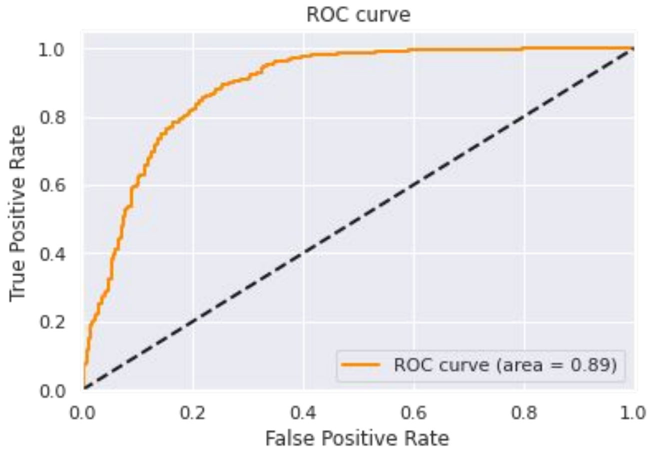
Residual Network

Training History

Accuracy & AUC

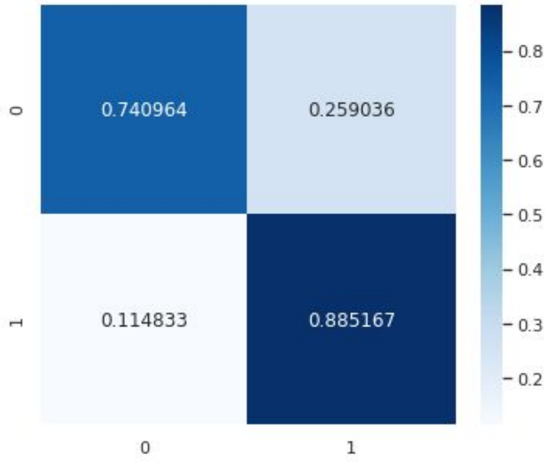
Confusion Matrix





The accuracy of the model is 0.807 while the AUC is 0.89.

<u>High TP & TN:</u> ResNet works well in identifying the slum images and non-slum images..



K-Means Clustering Analysis

<u>Training Data</u>

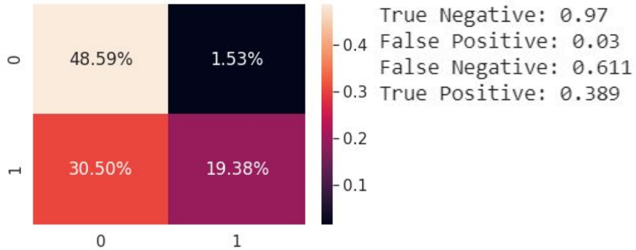
- Balanced data to 1045 slum-labeled images and the first 1050 non-slum labeled images.

Method

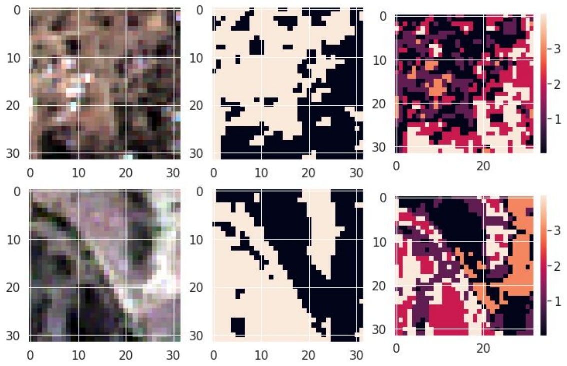
- Applied 2-cluster K-Means clustering analysis to the balanced dataset and calculated the confusion matrix to examine the classification accuracy.

- Classified one image from each label and applied 2-cluster and 5-cluster K-Means segmentation to see how the pixel distributed pattern for these two types of land uses.

Result



<u>High TN:</u> K-Means works well with identifying the non-slum images.
<u>Low TP & FP:</u> K-Means hardly classifies an image as a slum image.



The distribution of pixel value in slum images is more segmented; the pixel values distribution in non-slum images tends to be continuous and clustered.

K-Means is a feasible approach to differentiate slum and non-slum land uses by showing the pixel distributed patterns.

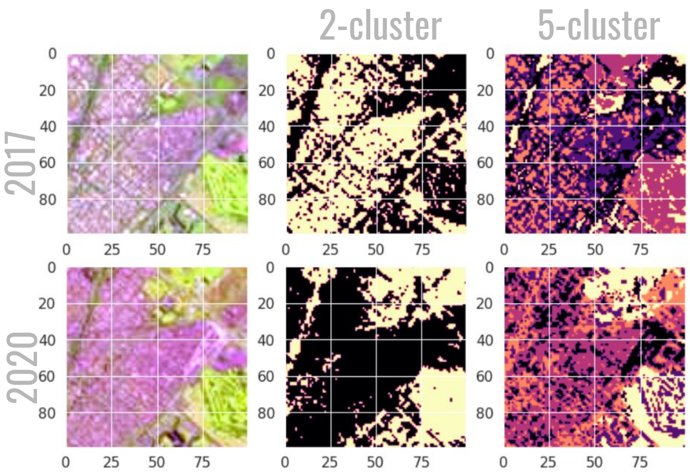
K-Means Clustering Analysis

Data for Application

Selected *Villa 20*, where it appeared as a slum in both CNN predicted results and the slum map. Applied 2-cluster and 5-cluster K-Means classifications to a Buenos Aires slum site in 2017 and 2020 images for two purposes:

- Explore the performance of K-Means on the mixed land uses images.
- Observe slum expanded or shrunk tendency over the three years

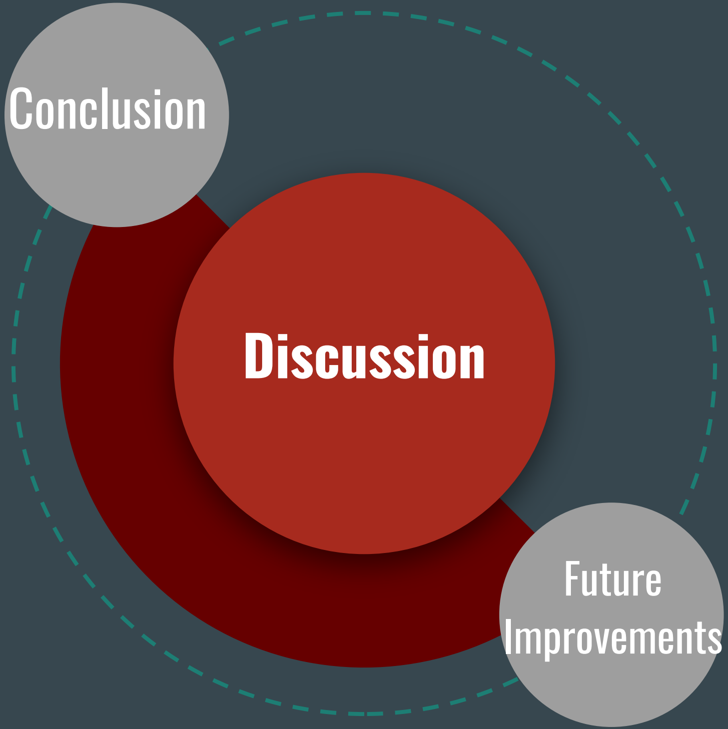
Process | **Result**



Only 2017 5-cluster image provides a fair identification of the slum area: the large proportion of purple pixels at the center of the image corresponds to slum location.

K-Means can barely identify slum out of other normal constructed land since slum is usually adjacent or embedded in the urban area. It works better when the dataset is binary (slum or non-slum images).

- Among K-Means, SVM, CNN, and ResNet, deep learning networks tend to predict well on the original dataset, and CNN has the best result in terms of slum identification in large satellite images.
- In general, based on the prediction produced by SVM and CNN, the number of urban slums / informal settlements increased from 2017 to 2020.



- To improve the model and avoid overfitting, we will consider:
  - Acquire higher resolution training data
  - Acquire more data
  - Get ground truth labels from other sources such as OSM
  - Use other methods of regularization