SatNet: A Low-Cost, Neural-Network based Algorithm Utilizing Publicly Available Data for Disease Hotspot Detection

Parkirat Sandhu
Academy of Engineering and Technology
Leesburg, Loudoun County
parkiratsandhu1@gmail.com
Orcid: 0009-0009-3980-6240

***Abstract*—The rapid spread of infectious diseases poses a significant global health challenge, requiring timely and accurate detection for effective intervention. Traditional disease detection services, such as the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO), play a crucial role in monitoring and responding to outbreaks. However, these services are largely inaccessible to people around the world due to their high costs and resource-intensive processes because they often rely on expensive sources of data. Fortunately, satellite images are a great alternative source of data as modern satellites can provide detailed images which clearly display a region's financial status and pollution levels: two key metrics in potential disease outbreaks. Therefore, this study aimed on developing a more affordable algorithm (SatNet) that utilizes publicly available satellite imagery to perform disease hotspot detection. The algorithm works by retrieving zoomed-in satellite images of the city inputted by the user and feeding these images into a novel, hybrid recursive convolutional neural network. This model, designed to classify regions within the images as low-income, high-income, or industrial areas, was trained and tested on a custom data set consisting of 7,448 images and was able to achieve a 94.872 training accuracy and 84.183 testing accuracy. The output of this model is then used to create a detailed heatmap for the city which clearly indicates the specific regions in most danger of disease outbreaks. Overall, the affordability and accessibility of SatNet will allow governments/organizations around the world to provide their people with the healthcare they need and significantly reduce the spread of diseases in an increasingly interconnected world.**

A. Inspiration

Infectious diseases are illnesses caused by bacteria, fungi, or viruses which enter a living body and multiply to create an infection that triggers a response of the immune system [1]. The severity of these diseases can vary greatly with some causing mild discomfort in the form of light fevers or body aches while others can be life-threatening with effects such as compromised immune systems or damage to vital organs [1]. However, what truly makes these illnesses so dangerous is their ability to spread rapidly through the air, bodily fluids, human-to-human contact, or even animal-to-human contact. Thus, infectious diseases have long plagued humanity, causing widespread devastation and posing significant challenges to public health systems worldwide [2]. Just recently, the emergence and rapid global spread of the novel coronavirus disease 2019 (COVID-19) has further highlighted the severe consequences of infectious diseases on modern society. During its 3-year, 3 months, and 5-day spell as a global pandemic, COVID-19 would infect over 760 million people worldwide and would be responsible for over 690 thousand deaths [3]. Furthermore, the increasingly interconnected global economy would suffer greatly as the pandemic would disrupt global supply chains leading to rapid inflation (as high as 8.73 percent according to the world bank) and high unemployment rates throughout the world [3]. These consequences have led many experts to argue for preventative measures and proactive action to reduce the spread of future infectious diseases [4]. As such, many have highlighted the need for more effective disease monitoring systems that can accurately predict the spread of illnesses within regions to ensure that proper action is taken [4]. This is further supported by recent findings which show that the recently implemented smart-phone based contact tracing was significant in reducing the spread of COVID-19 throughout the countries that utilized this technology [13]. Still, there are far more accurate and effective methods of limiting the spread of diseases. One such method is Disease Hotspot detection which is a process whose goal is to detect or predict specific regions within countries, states, or cities which are most likely to have the highest concentration or number of cases [14]. By identifying these locations, government agencies or private organizations can allocate resources to these areas and stop the outbreak from spreading further. Currently, many countries across the world have developed health programs such as the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) to develop disease hotspot detection systems, yet the operational costs of these programs make them inaccessible for a large percentage of the world's population.

B. Economic Dilemma

The American Center for Disease Control and Prevention (CDC), which is responsible for monitoring and preventing the spread of infectious diseases, had a budget of 10.675 billion dollars for the 2023 fiscal year [5]. With a population size of approximately 335 million, America will spend roughly 31.87

dollars per person on disease control in 2023 [5]. The total budget of 10.675 billion dollars is about 0.2 percent of the 4.8 trillion dollar revenue the U.S. government is projected to earn in 2023 [7]. On the other hand, if a developing country such as Pakistan was to spend the same 31.87 dollars per person for its population of roughly 234 million on disease control, the total budget would have to be 7.457 billion dollars. This would amount to approximately 19.83 percent of their 376 billion dollars projected government GDP for 2023 [6]. Case in point, many countries around the world, like Pakistan, simply can not afford traditional government-funded programs for disease control. This is largely because most programs, like the CDC, tend to use hospitalization records and government-funded national censuses to build disease hotspot detection services [8]. It is custom for doctors and medical professionals in the United States to record a patient's date of visit, illness, and treatment in sophisticated computer programs that supply this data to central federal databases [8]. However, most countries around the world do not possess the resources to provide computers within hospitals/clinics, afford collection software, or even create a federal medical database [9]. Therefore, the lack of traditional medical data limits these countries' health programs from being able to create any algorithms or tools that can protect their populations against disease outbreaks.

C. Satellite Imagery

Due to the high costs of traditional sources of data (as mentioned in section 1.2), the goal of this study was to identify and utilize alternatives for disease hotspot detection. The alternative source of data needed to be easily accessible, affordable, and equally as effective as hospitalization records or nationwide censuses. Our search led us across multiple sources of data such as pollution records, weather data, and population density information, but ultimately, satellite images were chosen. Modern satellite technology has allowed corporations and governments around the world to be able to image large parts of the planet using high-definition cameras. These images provide a level of detail that makes them a capable replacement for traditional numerical data used by the CDC and WHO. This is demonstrated in the diagram below:

Fig. 1. Suburb of Harrisburg, PA USA

Fig. 2. Slum in Rio De Janeiro, Brazil

Individual buildings, streets, and geographical features are clearly visible which provides a great insight into the financial and living conditions of the people in that area. Additionally, there exists a plethora of websites that grant users access to their satellite imagery database through APIs at low costs. Therefore, satellite images are easy to acquire, affordable, and can be used effectively with image classification models in order to develop disease hotspot detection software. As a result of all these factors, it was clear that satellite images were the ideal alternative source of data.

I. Method

A. Overview

The overarching purpose of this study was to create software that would accurately and affordably provide disease hotspot prediction for any city worldwide. Furthermore, it was intended for this software to be easy to use so that anyone around the world, regardless of their medical expertise, could effectively use this tool. Thus, much of the initial research period was spent on designing an software architecture that would fulfill these requirements. After Seventeen iterations the final design was established (shown in the figure below):

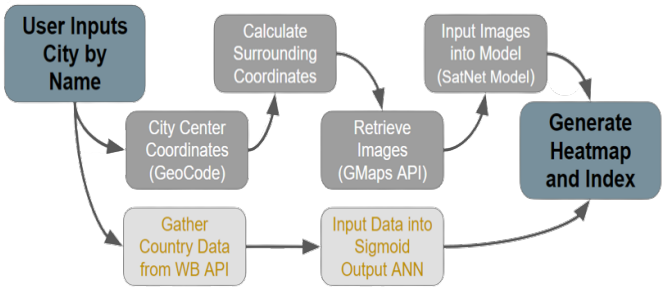

Fig. 3. Diagram of the proposed software's overall architecture.

The software starts by allowing a user to input the name of the city for which they wish to see disease hotspot detection analysis. Following this, the data for the city is gathered which includes zoomed-in satellite images that cover a 25-square-mile area of the city and its demographic/economic

information. The aforementioned regions captured in the satellite images are then classified based on financial, pollution, and quality-of-living factors while the demographic/economic data is used to produce an overall disease risk index for the city. Finally, a detailed visual is created which clearly indicates the regions within the city that are in most danger of a disease outbreak (identified using the satellite images and image classification model) while the overall risk index quantifies the level of risk the city in total faces compared to other cities in the world. Further details about all of the elements within the software's architecture diagram will be discussed in the following subsections.

B. User Input and Data Acquisition

The first thing the user is greeted by when opening the software is a prompt that allows them to type in the name of the city for which they would like to perform disease hotspot detection analysis. After the city name is acquired, the program then works on gathering the necessary data. For this, third-party databases would have to be accessed through special programs known as APIs (Application Programming Interfaces). APIs allow programs on a device to access foreign computers/databases to find and retrieve data available within them. The software developed for this study utilized APIs in two instances with the first being to collect satellite images of the city. For this task, the Google Maps API was used since it provides high-quality images, is low-cost (28,500 API calls for 200 dollars), and is easy to use because of its extensive documentation (available here). One of the downsides of the Google Maps API is that it does not accept city names within its call requests, rather it relies upon the geographic coordinate system to provide satellite images for desired locations. Therefore, the exact coordinates of a city are required to acquire its satellite images: a task accomplished by GeoCode. This is a Python library that takes in the name of the city, searches up the name on Google Search, then retrieves the coordinates of a central location within the city from the search result. However, this raised another issue. With just one pair of coordinates, only one Google Maps API call could be made resulting in the satellite image having to be very zoomed-out to cover the 25 square miles target zone.

Fig. 4. Satellite Image capturing 25 square mile area of Rio De Janeiro.

With the image being this zoomed-out, the individual features (buildings, roads, and terrain) are impossible to identify and distinguish. In turn, it would be impossible for an image classification model to be able to accurately classify the different regions within the image due to the lack of clarity. So, the solution to this problem was to gather multiple individual zoomed-in images of the various regions. Specifically, 400 images are collected, each capturing a 0.25-mile by 0.25-mile section of the city, which when arrayed together in a grid pattern create the intended 5-mile by 5-mile satellite image of the city. This is shown in the diagram below:

Fig. 5. Satellite Images City Grid Diagram

This solution, however, presented a problem of its own: each one of the 400 images would require its own set of coordinates for the Google Maps API calls. These coordinates could not be gathered using GeoCode since GeoCode only returned one pair of coordinates per city (the coordinates of a central location within the city). Thus each of the 400 pairs of coordinates (longitude and latitude) would have to be calculated individually. Calculating the latitude for each of the coordinates was rather easy as according to the United States Geographical Survey, 1 degree of latitudinal shift is approximately 69 miles. Thus, a 0.25-mile shift would equate to 0.00362 degrees. Meanwhile, due to the spherical shape of Earth, a longitudinal shift is not directly proportional to a change in miles. Rather, this relationship is dependent on the latitude at which the coordinate exists. For instance, a 1-degree change in longitude at a latitude of 30 degrees would equate to roughly 519 miles while the same change in longitude at a latitude of 60 degrees would equate to roughly 300 miles. In order to calculate the new longitudes, a special formula, developed by topographers at the Red Rock Canyon Conservation, was used which goes as follows:

$$\delta longitude = \frac{1}{cos(latitude \times \frac{\pi}{180}) \times 69} \times 0.155$$

Using these methods, the latitudes and longitudes of each of the 400 coordinate pairs are calculated in relation to the central coordinates obtained from GeoCode to form the 5-mile by 5-mile satellite image of the city. Following this, the Google Maps API is called with these new coordinate pairs to obtain

and store the satellite images for use in the image classification model.

The second instance of an API within this software was to collect demographic/economic data from the World Bank Database. This API was used to gather 12 key metrics for the city's home nation: GDP per capita (2023), net national income per capita (2023), morality from unsafe water (2022), Death by communicable diseases (2022), percent of the population with clean drinking water (2022), percent of the population that smokes (2022), percent of population with basic sanitation (2022), percent of population living under the global poverty line (2022), percent of population malnourished (2022), percent of population suffering food insecurity (2022), number of homicides per 1000 people (2022). Alongside this data, the results of the image classification model were used to generate the city's overall disease risk index (process explained in section 2.10).

C. Dataset

One of the greatest challenges during this study was finding a training data set for the image classification model. This is because this model was intended to take in the 400 satellite images of a city and then classify the areas within those individual images as either high-density low-income (slum), high-income mixed-density (rich), or heavy industrial (industry) regions. These classifications indicate the financial status, living conditions, and pollution levels of a specific area and provide the user with the necessary information to determine where disease outbreaks are most likely to occur. However, research involving the classification of this exact type has either not been done before or is not publicly available. In fact, there exists little in the space of utilizing urban satellite imagery for disease detection in general. Hence, it was impossible to find an adequate data set that could be used to train the image classification model. So, rather than trying to find a dataset, the effort was shifted to developing a novel dataset that could be used to train the image classification model. For this, 2 digital assistants were hired from Upwork who have had prior experience in data scraping, collection, and classification for research purposes. These assistants were tasked with providing a total of 22,500 coordinates: 7,500 from each of the 3 continental regions involved in this study (South America, Africa, and Southeast Asia) with this 7,500 comprising 2,500 slum areas, 2,500 rich areas, and 2,500 industrial areas from cities across the continent. All of the data was stored within a Google Sheets document (available here: Google Drive Link) with the first column containing the coordinates of the location, and the second column containing the classification of said coordinate (slum, rich, or industrial). Utilizing this approach to acquiring training data allowed for the dataset to be properly procured to this model and ensured that an adequate amount of data was available for the training.

D. Data Pre-Processing

For training purposes, 70 percent of the dataset (15,750 coordinates) was used as the training set while the remaining 30 percent (6,750 coordinates) was used as the testing set. This was accomplished by using the test-train-split method available in the Scikit-learn library. Once the data was split, the satellite images corresponding to all of the coordinates within the dataset were collected through 22,500 individual Google Maps API calls. These images, stored as .PNG files, are represented by a 640x640x3 array since the resolution of satellite images gathered using the Google Maps API was 640x640 with each individual pixel containing an array of 3 independent RGB values ranging from 0 to 255. Because of this, these images can be standardized by dividing all of the RGB values for each image by 255, thus allowing the image classification model to fit to the data during training easily. Following standardization, the images were uploaded to a Google Drive folder in order to be used within Google Colab for training purposes. However, since both the training and testing datasets were quite large, it was impractical to load up all the images within the Google Colab workspace due to memory constrictions. Therefore, an image generation class, named make-datagen, was created which stores image file names and classification and only loads up images when a new batch is required. The loaded-up images, for both training and testing sets, are then stored within a Pandas dataset, each consisting of 2 columns: the first one containing the 640x640x3 representing the image and the second containing the classification corresponding to the image. At the completion of this step, the entire Pandas dataset, which contains 1 batch, is inputted into the model either for training or testing purposes.

E. Image Classification Model

For the purposes of this study, a novel hybrid convolution neural network (CNN) was developed for classifying the regions within satellite images as high-density low-income (slum), high-income mixed-density (rich), or heavy industrial (industry) regions. This model combined a ResNet50V2 (ResNet), InceptionV3 (InceptionNet), and MobileNetV3 (MobileNet), each of them pre-trained on the ImageNet dataset, to create a highly accurate image classification model. On their own, each one of these models is very powerful with all of them scoring within the top 10 percent of highest accuracy on the ImageNet dataset amongst other CNNs. However, due to the complicated yet varied street patterns and building designs in each satellite image, a superior model was needed that could accurately identify these patterns and classify the images. Thus the 3 aforementioned models were used in conjunction in order to combine their advantages and allow for more accurate image classification. In order to accomplish this concatenation of three models, the following model architecture was used:

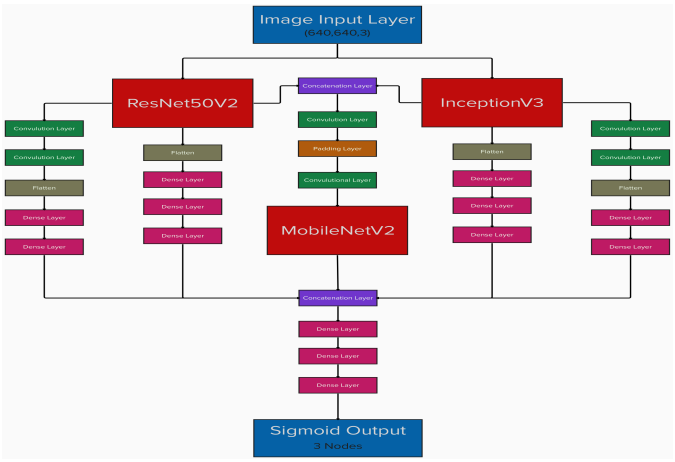

Fig. 6. Image Classification Model Architecture

The model starts off with an individual input layer that accepts arrays of size 640x640x3, the same dimensions as the arrays which represent an image. This input array then feeds into the first layers of the ResNet and the Inception, each of whose tops (input layers) have already been removed during their initialization. Around a quarter of the way through both models, outputs from certain layers were collected, resized, and concatenated to serve as input values for the MobileNet model. Specifically, the outputs of the "mixed1" layer in the InceptionNet and the "conv3-block1-out" layer in the ResNet were taken. Then, since the dimensions of the output from both layers were different ("mixed1": 77x77x218, "conv3-block1-out": 80x80x512), the output from the "mixed1" layer was sent through a 2D padding layer which added 3 rows and 3 columns of zeros to its array. Following this, the data from both layers were concatenated together to create an output of size 80x80x800. This was then connected to 2 more 1x1 2D convolutional layers which output arrays of dimensions 80x80x1024 and 80x80x3 respectively. Finally, the latter of the aforementioned convolutional layer was connected to the MobileNet model. Additionally, connections from the middle of both the ResNet and InceptionNet were created to serve as residual layers for the overall model. The outputs from the "conv4-block3-out" layer in the ResNet and the "mixed5" layer in the InceptionNet were passed through two 2x2 2D convolutional layers, 1 flattening layer, and 2 dense layers with the latter outputting a 1x64 sized array. These outputs were to be concatenated with the outputs of the 3 models (ResNet, InceptionNet, and MobileNet), but since each model's output size was different, they had to be reshaped through a series of layers. The outputs of each of the models were passed through multiple dense layers until an output size of 1x64 was achieved for all of them. At this point, the output from the 2 residual layers, ResNet, InceptionNet, and MobileNet were concatenated together to create a 1x320-sized array. This was then passed through 4 more dense layers with the last one (final output layer) consisting of 3 neurons and a sigmoid activation function. Overall, this model combines 3 powerful pre-existing models in a residual architecture for optimal image classification performance.

F. ResNet50V2

ResNet50, a convolutional neural network (CNN) architecture, has garnered notable success by achieving a top-1 accuracy of 0.749 and a top-5 accuracy of 0.921 on the ImageNet database [15]. Its effectiveness stems from its utilization of a residual architecture in which the output from blocks of convolutional layers is passed into the following blocks. This innovative approach effectively addresses the issue of vanishing gradients, an issue that leads to early layers within the network not properly being trained due to diminishing gradients. By mitigating this problem, ResNet enables the training of deeper networks with more accurate results. ResNet50, specifically, consists of 50 total layers with 1 input layer, 48 convolutional/pooling/BatchNormilization/activation layers, and 1 output layer. The ResNet50 architecture is shown in the figure below:

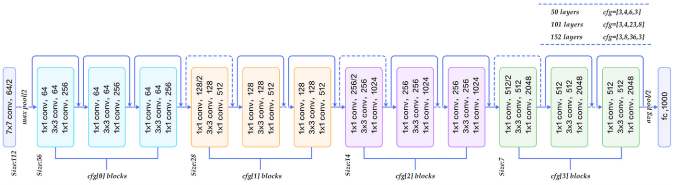

Fig. 7. Diagram of ResNet50V2 architecture [11]

G. InceptionV3

InceptionV3 is a powerful deep convolution neural network that achieved a top-1 accuracy of 0.779 and a top-5 accuracy of 0.937 on the ImageNet dataset [15]. Its effectiveness stems from its use of large blocks of convolutional layers with differing kernel sizes which allows the network to train on a multitude of filter sizes without manual changes. Consequently, the network is able to determine which filter size is optimal for the dataset during training and is able to adjust its weights in order to achieve the highest degree of accuracy. The InceptionV3 architecture is shown in the figure below:

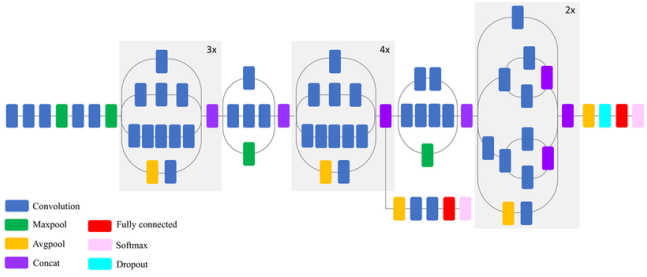

Fig. 8. Diagram of InceptionV3 architecture [12]

H. MobileNetV2

MobileNetV2 is a deep convolutional neural network that achieved a top-1 accuracy of 0.747 and a top-5 accuracy of 0.908 [15]. While these accuracies are not the highest when compared to the other networks mentioned above, the specialty of MobileNetV2 is its efficiency and small foot footprint. The MobileNetV2 also utilizes a residual architecture, just like the ResNet, but it uses significantly fewer layers and residual blocks than its counterparts. For example, the ResNet50 contains over 23 million trainable parameters, InceptionV2 contains 56 million trainable parameters, and MobileNetV2 only contains 3.4 million parameters. It is able to achieve this efficiency by rapidly downsizing input images and then using the smaller dimensions with small filtered convolutional layers to reduce the number of parameters in the model. The MobileNetV2 architecture is shown below:

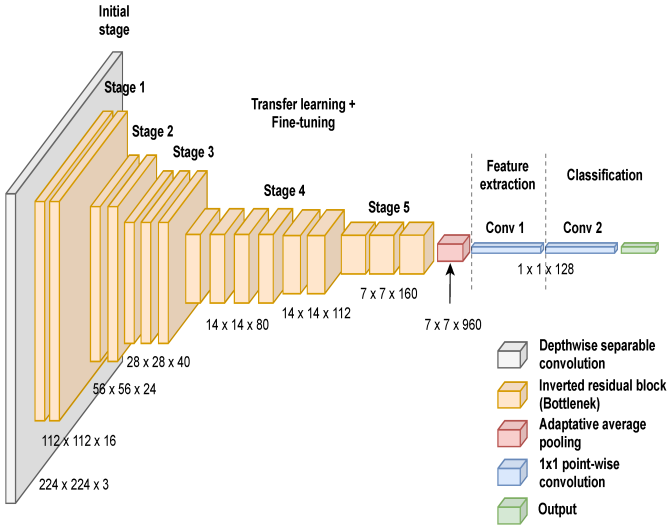

Fig. 9. Diagram of MobileNetV2 architecture [10]

I. Heat Map Generation

The resulting output from passing a satellite image through the image classification model is a 1x3 array containing probability values. These values are received from the sigmoid activation function in the final layer of the model and each value indicates the likelihood (in decimal values) that a satellite image is either a high-density low-income (slum), high-income mixed-density (rich), or heavy industrial (industry) region. While these arrays of probability values provide great insight, they are difficult to interpret which leads to a need for a simple yet comprehensive visual to better organize the information. For this, a Python library called MatPlotLib was used: a tool used for developing complex visuals, 3D models, and graphs. In the case of this study, MatPlotLib was used to create a detailed heatmap that clearly indicates the regions within the city with the highest chance of a disease outbreak. This is done by first passing all of the satellite images related to a city through the image classification model and storing their output arrays into a 2D master array. After this, each of the 3 values within the individual output arrays is multiplied by 255, resulting in a RGB array. Since the individual arrays were stored in a 2D array, their location within the array corresponds to their images' location within the city. Thus the master array is directly fed into the matplotlib function, which accepts 2D arrays as a parameter, in order to create the heatmap (shown below).

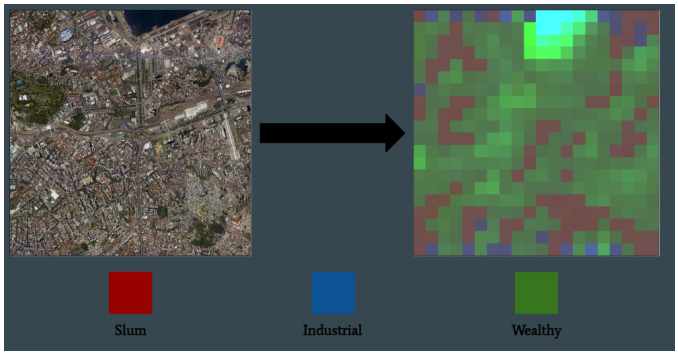

Fig. 10. Heatmap of City Disease Hotspot Detection

This heatmap consists of multiple squares, each of which represents the output from its corresponding satellite image being passed through the image classification model. The color of each square is determined by the output array after it is multiplied by 255 as it serves as a RGB value. A more reddish-colored square indicates a close resemblance to a slum, a greenish square indicates a wealthier district, and a bluish square indicates heavy industry. When all squares of the heatmap are put together it mirrors the large overall satellite image of the city. Therefore, when put side-by-side, the overall satellite image of the city and its heatmap allows a user to easily identify specific regions within a city that are most likely to experience a disease outbreak.

J. Disease Risk Index

As important as it is to identify specific disease hotspots within cities, it is also crucial to provide a numerical indicator of a city's overall susceptibility to diseases. For this, a disease risk index was generated and displayed above the heatmap visual. This index is calculated by using the demographic/economic data from the world bank database as well as the number of satellite images identified as slum, wealthy, and industrial regions by the image classification model. To process this data, a basic, 5-layer regression deep neural network with an input layer that accepts 1x15 sized arrays, 3 dense layers (16, 32, and 8 nodes respectively), and 1 output layer (no activation function). The model was trained on a custom dataset that labeled assigned the highest possible index (1.000) to the 10 cities with the most recent major disease outbreaks as recorded by the CDC. Each city on this list was passed through the overall software to collect its demographic/economic metrics and generate its heatmap which provided the necessary data to train the model. At this point, the Disease Risk Index generator was incorporated into

the overall software as the final step which occurs just before the visual is shown to the user. All the data is collected from the image classification model and the World Bank Database to create an index as shown in the figure below:

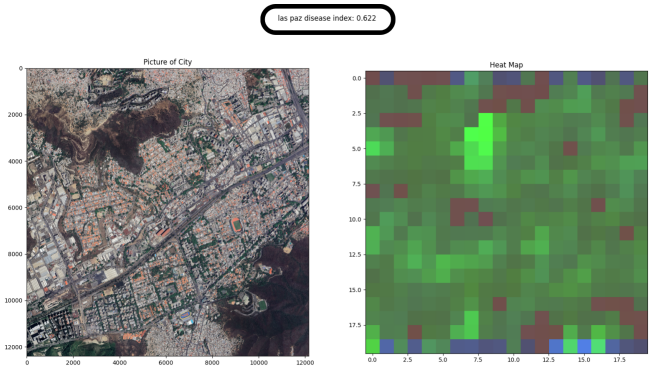

Fig. 11. Disease Risk Index for City of Las Paz, Bolivia

II. Results

A. Accuracy

Due to the nature of this study, it is very difficult to quantify the accuracy of the SatNet tool. This is because the whole purpose of this software was to create software that allows users to easily identify disease hotspots within cities. For this task, the image classification model was used to classify the regions in satellite images based on their financial and demographic metrics. However, there exists little to no data which catalogs the economic and population metrics of specific coordinate locations within cities. Moreover, the little economic and population data that does exist is private and very difficult to access. Thus it is impossible to provide an exact statistic regarding the accuracy of the software in its real-world application. However, the results from the training and testing datasets due provide a quantitative value. In terms of the satellite image classification model, the network was able to achieve a 95.56 percent training accuracy and a testing accuracy of 87.42 percent on the dataset mentioned in section 2.3. In comparison, 3 popular models (ResNet50V2, InceptionV3, and Vgg16) were trained on the same dataset to serve as controls and their results are shown below.

TABLE I
Training and Testing Accuracies Values of Different Models on the Dataset Created for this Study

Model	***Training Accuracy***	***Testing Accuracy***
SatNet Model	94.97%	84.18%
ResNet50V2	88.29%	81.44%
InceptionV3	86.81%	72.49%
VGG16	62.29%	54.19%

It can be seen that the image classification model designed for this study outperforms other popular CNN architectures when applied to this dataset. On the other hand, the model created for the disease risk index generation did not show the same promise. Due to the limited data within the dataset, the model was overtrained as it was able to achieve a 100 percent training accuracy in less than 10 epochs. This is a clear indication that the model was overtrained onto the data as attaining a 100 percent training accuracy should not be possible.

B. Project Cost

One of the primary goals of this study was to ensure that any solution/software developed for disease hotspot detection was low-cost. In all, this study was able to accomplish this goal as the overall cost of the project was 680 dollars. This was split into 2 main expenses: 480 dollars paid to the digital assistants that helped create the dataset and 200 dollars for 4 months of Google Colab Pro membership. In comparison, the software developed in this study costs only 0.0000063700234192 percent of the CDC's 10.675 billion dollars proposed budget for the 2023 fiscal year. However, due to the dependency on certain third-party APIs and software, there is a small fee each time a user requests an analysis on a particular city. Specifically, the Google Maps API charges 440 dollars for 250,000 call requests, and as each city requires 400 total satellite images, the charge per city comes out to roughly 70 cents. This means that the cost of developing the software and performing city disease hotspot analysis 1000 different times for 1 year would cost just 9182 dollars: far lower than the annual budget of most government agencies. Therefore, this software was successfully able to achieve its goal of affordability as the relatively low development and use prices will ensure that anyone could utilize this tool.

III. Conclusion

A. Impact

With the world becoming increasingly interconnected, the threat of communicable diseases and their potential consequences has quickly become one of humanity's greatest obstacles. Just recently, the COVID-19 pandemic wreaked havoc across the planet as millions lost their lives and the global economy was greatly disrupted. Even today, over 3 years from the start of the pandemic, its effects can still be felt to a great degree with rising inflation, unemployment, and homelessness rates ravaging countries worldwide. Unfortunately, future pandemics will only cause more devastation as the growth of intercontinental travel and international trade greatly increases global connectivity and interdependence. All of this is only amplified by the fact that many regions of the world will continue to lack adequate early disease detection and disease hotspot detection due to the high price of traditional solutions. While the software developed during this study will not entirely eradicate the threat of infectious diseases, it could play a crucial role in significantly reducing the danger humanity faces from future outbreaks. By providing an affordable disease hotspot detection tool, countries and non-profit organizations worldwide can now begin to protect their people, many of whom were previously vulnerable. Furthermore, this software will allow for proactive action and will ensure that critical

medical resources, such as medications, vaccines, personal protective equipment, etc., are sent to regions that need them the most. Ultimately, the satellite images retrieval algorithm, image classification model, disease risk index generator, and detailed heatmap developed in this study provide a foundation for future researchers to greatly improve upon. Considering its effectiveness, despite the budgetary and time constraints of this study, it is likely that with greater resources, improved datasets, and increased manpower, SatNet can be significantly improved and become far better at protecting humanity.

B. Flaws

Throughout this study, a few compromises were made, due to budgetary or time-related factors, which should be addressed in any future research regarding this topic. One example of this is the platform used to attain satellite images. Because Google Maps was designed as a navigational tool, its images are not updated very often as the long duration for construction/development allows Google to retain outdated maps. This is especially prevalent within major cities which hardly see much short-term development due to the lack of available space. As a result, there are many locations across the planet for which the most recent images that Google Maps has are from 2017. This is a major issue when trying to provide up-to-date disease hotspot analysis as development in slums, pollution, landfills, etc. can not be accounted for due to the outdated satellite imagery. Fortunately, there do exist platforms that provide daily, worldwide satellite images but they were not used in this study due to their complicated documentation and lack of compatibility with Python-based programs. In addition to this, the Google Maps API was far more affordable, especially for large amounts of call requests. Another area of improvement in this study was the data used for training/testing the image classification model. Although the solution of relying on experienced digital assistants for data collection was innovative and effective, there is undoubtedly a level of uncertainty regarding the validity of this data. Even though the dataset was thoroughly combed to ensure it met the standards, the fact that it was not developed by a reputable researcher or institute brings its reliability into question. This could be mitigated by trying to contact researchers or institutions that may have private data which could be used for this project or could validate the dataset created during this study. However, both of these are time-intensive processes that could take months or years to reap results during which time thousands of people could suffer from the spread of infectious diseases. The final major area of improvement for this study would be the inefficiency of the image classification model. On average, the model takes 7 minutes to classify all 400 images required for the analysis of each city. A major reason behind this is the large image size as each has a 640x640 resolution; this equates to 409,600 individual pixels with each having 3 RGB values. Thus the sheer number of parameters that the model must input and process significantly increases classification times. However, because bigger satellite images are able to capture larger parts of the city, it is impossible to reduce their resolutions since a smaller area of the city would be analyzed. This can be countered by reducing the zoom of each image as a more zoomed-out image could capture the same location with a smaller image, but this leads to the problem of images losing details that are critical for the classification process. Ultimately, the best way to get around this problem would require a combination of solutions. This includes creating a more efficient image classification model (3rd party models from Google's Deep Mind or Meta AI) and finding the perfect zoom amount that retains the necessary details in smaller image sizes. Aside from the aforementioned, there are many smaller areas of improvement that should be addressed in future studies to create more efficient, accurate, and effective disease hotspot detection software.

C. Future Plans

The upcoming additions to this software all revolve around 1 goal: accessibility. In its current state, this software is not accessible to anyone outside the members of this study, and thus governments, organizations, or even regular people are not able to utilize it. Therefore, the ultimate objective is to make improvements and upgrades which allow anyone around the world to use this software for disease hotspot detection. For this, the current plan is to create a website, hosted on Amazon Web services, which will allow anyone in the world to utilize the software. Essentially, users will be able to access the website through the internet, type in the name of the city for which they wish to perform disease hotspot analysis, and receive the heatmap/disease risk index after the software is executed on a foreign server. However, this approach creates 2 major problems. The first is in regard to the aforementioned issue of lengthy analysis time. It takes, on average, just over 10 minutes between the time the user inputs the name of a city and the generation of its heatmap. This type of long wait time is unacceptable for websites and would prove to be a deterrent as many users would not wish to wait for such long periods of time for analysis. Furthermore, the sheer computing power that SatNet (available here) requires to perform its analysis makes it extremely expensive to host on 3rd party servers (such as AWS) as many charge based on the intensity of the program. Even if a few hundred people were to request an analysis per second, the significant computing power required to fulfill these requests would lead to high operational expenses (roughly 307 dollars per second). This completely defeats the purpose of this study as increased operation expenses would force users to pay much more per analysis, making it far more inaccessible for the average person. These issues could be remedied by using a more efficient image classification model (mentioned in section 4.2) and using improved data retrieval methods, such as asynchronous APIs. However, each solution presents its own unique downsides which must be further studied for optimization. Thus, despite the fact that a general plan for the future development of SatNet does exist, there are many details that still must be figured out and issues that must be addressed.

REFERENCES

[1] CDC, "Who We Are — NCEZID — CDC," www.cdc.gov, Jul. 07, 2020. https://www.cdc.gov/ncezid/who-we-are/index.html#:~:text=Infectious%20diseases%20are%20illnesses%20caused(accessedJul.28, 2023)

[2] R. E. Baker et al., "Infectious disease in an era of global change," Nature Reviews Microbiology, vol. 20, no. 20, Oct. 2021, doi: https://doi.org/10.1038/s41579-021-00639-z.

[3] S. Silva, E. Goosby, and M. J. A. Reid, "Assessing the impact of one million COVID-19 deaths in America: economic and life expectancy losses," Scientific Reports, vol. 13, no. 1, p. 3065, Feb. 2023, doi: https://doi.org/10.1038/s41598-023-30077-1.

[4] T. Saito et al., "Proactive Engagement of the Expert Meeting in Managing the Early Phase of the COVID-19 Epidemic, Japan, February–June 2020 - Volume 27, Number 10—October 2021 - Emerging Infectious Diseases journal - CDC," wwwnc.cdc.gov, vol. 27, no. 10, Oct. 2021, doi: https://doi.org/10.3201/eid2710.204685.

[5] CDC, "CDC Newsroom," CDC, Jan. 01, 2016. https://www.cdc.gov/media/releases/2022/s0328-2023-budget.html#:~:text=The%20Centers%20for%20Disease%20Control (accessed Jul. 28, 2023).

[6] A. Shahzad, A. Shahid, A. Shahzad, and A. Shahid, "Pakistan set for 0.29% GDP growth in FY23," Reuters, Jun. 08, 2023. Accessed: Jul. 28, 2023. [Online]. Available: https://www.reuters.com/markets/asia/pakistan-gdp-growth-seen-029-fy23-economic-survey-2023-06-08/

[7] Congressional Budget Office, "The Budget and Economic Outlook: 2023 to 2033 — Congressional Budget Office," www.cbo.gov, Feb. 08, 2023. https://www.cbo.gov/publication/58946 (accessed Jul. 28, 2023).

[8] F. P. Havers, "Laboratory-Confirmed COVID-19–Associated Hospitalizations Among Adults During SARS-CoV-2 Omicron BA.2 Variant Predominance — COVID-19–Associated Hospitalization Surveillance Network, 14 States, June 20, 2021–May 31, 2022," MMWR. Morbidity and Mortality Weekly Report, vol. 71, no. 34, 2022, Available: https://www.cdc.gov/mmwr/volumes/71/wr/mm7134a3.htm

[9] K. Kelland, "One Billion People Cannot Afford healthcare: WHO," Reuters, Nov. 22, 2010. Accessed: Jul. 31, 2023. [Online]. Available: https://www.reuters.com/article/us-who-finance/one-billion-people-cannot-afford-healthcare-who-idINTRE6AL1GV20101122

[10] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery," Remote Sensing, vol. 10, no. 7, p. 1119, Jul. 2018, Accessed: Nov. 26, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/10/7/1119

[11] A. Rastogi, "ResNet50," Medium, Mar. 14, 2022. https://blog.devgenius.io/resnet50-6b42934db431 (accessed Jul. 31, 2023).

[12] M. Abd Elaziz, A. Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba, and M. Ahmadein, "Boosting COVID-19 Image Classification Using MobileNetV3 and Aquila Optimizer Algorithm," Entropy, vol. 23, no. 11, p. 1383, Oct. 2021, doi: https://doi.org/10.3390/e23111383.

[13] E. Hernández-Orallo, P. Manzoni, C. T. Calafate, and J. Cano, "Evaluating How Smartphone Contact Tracing Technology Can Reduce the Spread of Infectious Diseases: The Case of COVID-19," IEEE Access, vol. 8, pp. 99083–99097, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2998042.

[14] J. Lessler, A. S. Azman, H. S. McKay, and S. M. Moore, "What is a Hotspot Anyway?," The American Journal of Tropical Medicine and Hygiene, vol. 96, no. 6, pp. 1270–1273, Jun. 2017, doi: https://doi.org/10.4269/ajtmh.16-0427.

[15] S. Mattapalli, R. Athavale, T. Jefferson, and H. School, "ALLNet: A Hybrid Convolutional Neural Network to Improve Diagnosis of Acute Lymphocytic Leukemia (ALL) in White Blood Cells." Accessed: Aug. 01, 2023. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2108/2108.08195.pdf