University of Nairobi

School of Computing and Informatics

# CONVOLUTIONAL NEURAL NETWORK FOR PREDICTING POVERTY USING SATELLITE IMAGERY

**Author**

Brian Mogambi

**Supervisor**

Prof. Robert Oboko

RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE DEGREE IN COMPUTATIONAL INTELLIGENCE, UNIVERSITY OF NAIROBI.

August 2021.

# Declaration

I declare that this research project is my original work and that it has not been submitted, in whole or in part, to any other university.

**Name:** Brian Mogambi **Registration Number:** P52/34066/2019

**Signature:** .................... **Date:** 21/08/2021

This research project has been submitted in partial fulfillment of the requirements for the award of the degree of Master of Science in Computational Intelligence at the University of Nairobi with my approval as the university supervisor.

Prof. Robert Oboko
School of Computing and Informatics
University of Nairobi

**Signature:** .................... **Date:**....................................

# Acknowledgments

This project would not have been possible without the support of many people. Many thanks to my supervisor, Prof. Robert Oboko, and my committee members, Prof. Agnes Wausi, Prof. Elisha Opiyo, and Ms. Pauline Wangunyu, for their unwavering guidance throughout the project.

Lastly, my family deserves endless gratitude. To my father, mother, and brother, I am grateful for the unconditional and loving support.

# Abstract

Eradication of poverty in all its forms worldwide by 2030 is the first of 17 Sustainable Development Goals outlined by the United Nations. In Africa, where the bulk of the world's poorest people live, national household surveys are used to collect data on poverty. It is difficult to obtain accurate and timely information due to the significant resource costs associated with conducting such surveys. Public access to abundant data sources such as daytime satellite images and nighttime lights in conjunction with advances in computer vision provides a feasible solution to the data scarcity problem. The purpose of this research is to follow previous research works in utilising machine learning techniques to process daytime satellite images and nighttime lights in order to predict the distribution of poverty at the village level in three African countries using more advanced technologies and up-to-date data. The procedure involves designing and training a Convolutional Neural Network, which is then used to extract poverty indicators from daytime satellite images. The extracted features are mapped to poverty statistics and used in a regression model to estimate poverty. In the initial nighttime light prediction task, the model attained an accuracy of 75.29% on the training data and 77.97% on the validation data over 81,000 iterations. In the target task of poverty estimation, the model explained 18% to 43% of the variation in average household expenditures in each country. The results demonstrate that the transfer learning technique is generally applicable to forecasting poverty in other countries, although its effectiveness is highly dependent on the hyperparameters used to tune the algorithm. Additionally, the technique does not easily transfer to other poverty indicators such as child mortality and levels of education.

***Keywords:*** machine learning, poverty, convolutional neural network, satellite images

# Table of Contents

# List of Figures

# List of Tables

# Abbreviations

**ANN** Artificial Neural Network. 5

**CE** Cross-Entropy. 24

**CNN** Convolutional Neural Network. iii, 2–4, 6–9, 11, 12, 14, 15, 20, 21, 24, 29

**CRISP-DM** CRoss Industry Standard Process for Data Mining. 14

**CV** Cross-Validation. 11, 15, 22, 26

**DHS** Demographic and Health Survey. 11

**DMSP** Defense Meteorological Satellite Program. 17

**DNB** Day/Night Band. 17

**DNN** Deep Neural Network. 5, 6

**EOG** Earth Observation Group. 17

**GCP** Google Cloud Platform. 22

**GDP** Gross Domestic Product. 11

**ILSVRC** ImageNet Large Scale Visual Recognition Challenge. 8, 20

**JPSS** Joint Polar-orbiting Satellite System. 17

**LASSO** Least Absolute Shrinkage and Selection Operator. 21

**LSMS** Living Standards Measurement Study. viii, 4, 9, 15–19, 24–27

**OLS** Operational Linescan System. 17

**ReLU** Rectified Linear Unit. 7

**USD** United States Dollar. 16

**VGG** Visual Geometry Group. 8, 20, 21, 24

**VIIRS** Visible Infrared Imaging Radiometer Suite. 17

# Chapter 1

# Introduction

## 1.1 Background

Poverty is defined as a lack of sufficient resources to meet one's basic needs (Okalow, 2021). Poverty mapping is the process of determining the spatial distribution of inequality and poverty in a given area. Measuring poverty is crucial in the implementation and evaluation of poverty eradication programs. Traditionally, poverty statistics are collected from national household surveys. There is a lack of sufficient poverty data in developing countries, owing to the high costs of conducting such surveys. This data gap is a critical barrier to poverty alleviation.

In Africa, this data gap is huge. Although Africa is home to some of the world's poorest people, many African countries do not publish poverty data (World Bank, 2015). This lack of reliable data at lower administrative levels in African countries is a key impediment to long-term development that could help policymakers and donors effectively implement poverty eradication programs (Xie et al., 2016). Advancements in information technology, remote sensing, and high-resolution satellite imagery could provide an alternative method of estimating poverty data. Commercial satellite sources will be able to provide daily global coverage at a sub-meter resolution at a fraction of the cost within ten years (Murthy et al., 2014). This big data could provide a bridge for developing countries to cross the data gap. However, this data is highly unstructured. Satellite imagery does not explicitly contain labels categorising the wealth of the area depicted without intelligent processing. Therefore, machine learning approaches are needed to extract such insights from the unstructured data.

A Convolutional Neural Network (CNN) is a type of deep neural network that is used to process data in a grid pattern. Deep learning methods used in large-scale datasets like ImageNet have revolutionised the field of computer vision in the last five years, resulting in substantial advances in fundamental tasks like object recognition (Russakovsky et al., 2014). The combination of CNN's trained against satellite imagery enables the creation of frequently updated and accurate poverty maps. Many research teams have explored the capabilities of CNN's trained against satellite imagery to estimate poverty. (Jean et al., 2016) used a transfer learning method by using the CNN's last layer as a feature extractor, while (Engstrom et al., 2017) used intermediate features (cars, roofs, plants) identified from computer vision to measure poverty. In contrast to current methods, CNN's can provide precise poverty estimates by using publicly accessible data.

## 1.2 Problem Statement

While the quantity and quality of economic data available in countries has increased in recent years, statistics on important indicators of economic progress remain scarce for a huge proportion of the developing world (Jütting & Donnell, 2017). African countries rely heavily on survey data to make decisions and plans, especially when it comes to measuring and monitoring poverty. Conducting surveys is time-consuming, labour intensive, and expensive, hence not frequently conducted. One of the most challenging obstacles to overcome in the fight against poverty is this lack of data.

There is a need to leverage innovative machine learning tools in the field of computer vision in combination with publicly available and inexpensive satellite data to close this data gap and consequently transform efforts to estimate and monitor poverty in African countries.

## 1.3 Research Objectives

### 1.3.1 General Objective

To implement a CNN model for predicting poverty using satellite imagery of three African countries.

### 1.3.2 Specific Objectives

The specific objectives of the research are:

1. To implement a CNN model that predicts nighttime light intensity from daytime satellite imagery.

2. To implement a regression model that predicts poverty from extracted features of the satellite imagery.

3. To evaluate the performance of the CNN model in image classification.

4. To evaluate the performance of the regression model in poverty prediction.

## 1.4 Significance

In the developing world, reliable statistics on economic livelihoods are limited, hindering efforts to study these outcomes and devise policies that improve them (Xie et al., 2016). This project provides a method for estimating poverty using satellite images that are publicly available and inexpensive. Using satellite data from three African countries, the project demonstrates how a CNN can be trained to extract relevant image features that correlate with poverty indices.

This machine learning method has the potential to improve efforts in tracking and monitoring poverty programs in developing countries. It demonstrates how advanced machine learning techniques may be used in a situation with minimal training data, indicating that they could be used in a variety of scientific disciplines (Jean et al., 2016).

## 1.5 Assumptions

The following assumption was made in this research:

1. The nighttime satellite imagery are from 2015, while the daytime satellite imagery for Malawi are from 2016. It is assumed that the target areas have not changed considerably during that period.

## 1.6 Limitations

1. The LSMS survey coordinates are reported as an average in a given region to maintain the anonymity of the respondents. This introduces a measurement error in the response variable.

2. Due to limited data on consumption expenditure for training the model, transfer learning will be used to train the CNN models using nighttime lights as a data-rich proxy for economic outcomes.

## 1.7 Scope of Study

This research focuses on nighttime lights intensity and consumption expenditure as poverty indices. However, the prototype solution presented can be applied in other classification tasks using different poverty indices.

# Chapter 2

# Literature Review

## 2.1 Introduction

This chapter introduces machine learning concepts and their corresponding mathematical formulations. They will be invoked later on in the applications. It additionally introduces the transfer learning approach used to train the model, related research works, research gaps, and finally illustrates a conceptual framework.

## 2.2 Theoretical Framework

### 2.2.1 Deep Learning

Deep learning is a branch of machine learning focused on Artificial Neural Network (ANN), algorithms based on the brain's functioning (Brownlee, 2020). Deep learning algorithms consist of a sequence of layers connected to each other and whose components are processing units referred to as neurons. There are three categories of layers: input layer, hidden layers, and output layer. For multi-layer neural networks, also known as feed-forward neural networks, layers are fully connected, and each neuron receives the information from the previous layer, processes it, and sends the integrated information to neurons of the subsequent layer (Nielsen, 2019). This process is referred to as forward propagation. The outcome from the output layer is the model prediction. A neural network with many hidden layers is referred to as a Deep Neural Network (DNN). Figure 2.1 displays the architecture of a Feedforward neural network.

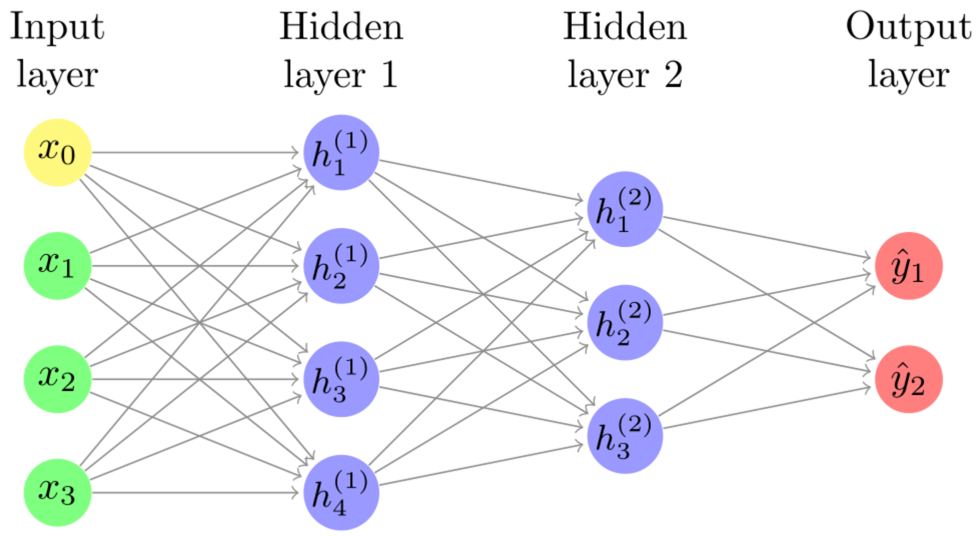



***Figure 2.1:*** *Feedforward neural network*

DNN's that utilise fully connected layers use many parameters which make the model hard to train, but also in the case of images, breaking an image into a vector of pixels can lead to the loss of certain neighbourhood properties (Bengio & Glorot, 2010). The next section introduces CNN's. They are a variant of DNN's which take two-dimensional input feature maps. They utilise convolutional operations and shared weights to summarise the input feature maps into vectors of feature representations. CNN's are widely used in image analysis due to their ability to take in a large number of inputs and reduce the dimensions systematically, managing the number of parameters to be trained and improving overall performance.

### 2.2.2 Convolutional Neural Network

Convolutional Neural Networks are neural networks specifically designed for data that has a known grid-like topology structure and involve convolution operations (Goodfellow et al., 2016). Convolution operations encode translational invariance, an important aspect of image features (Bouvrie, 2006). The operations involve the multiplication of an array of weights and input data. Convolutions allow an efficient implementation of the forward propagation and a remarkable decrease in the network's number of parameters. CNN models have layers of convolution operations, where the initial layers of the CNN learn image features such as edges, while the last layers learn features such as objects (Zeiler & Fergus, 2013).

There are three main categories of layers in convolutional networks: convolutional, pooling, and fully connected layers. Figure 2.2 illustrates the sequence of the layers in a typical CNN. The convolutional and fully connected layers have parameters to learn while pooling does not.

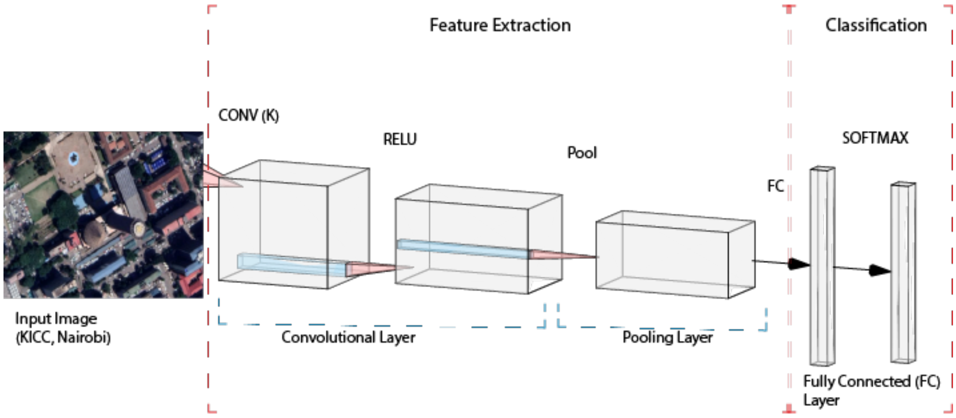



***Figure 2.2:*** *Convolutional Neural Network Structure*

### Convolutional Layer

Feature extraction is performed by the convolutional layer, which generally comprises of a combination of linear and nonlinear processes, such as convolution and activation functions (Yamashita et al., 2018). The application of a small array of numbers known as a filter over the input, which is an array of numbers known as a tensor, is known as convolution. Each kernel element's element-wise product with the input tensor is computed and totalled to produce the output value known as a feature map (Yamashita et al., 2018). A nonlinear activation function, such as a Rectified Linear Unit (ReLU), is then applied to the outputs of the linear convolution operation.

### Pooling Layer

The pooling layer involves applying a pooling function to the convolutional layer output, using a window with hyperparameters of stride, padding, and window sizes (Yamashita et al., 2018). This technique lowers the output's spatial dimensions while maintaining key input image data. The most often used pooling procedure is max pooling, which extracts features from inputs and outputs the maximum value in the feature (Yamashita et al., 2018).

#### Fully Connected Layer

Convolution or pooling layer outputs are flattened and connected to dense layers (Goodfellow et al., 2016). The fully connected layer has the same number of outputs as the number of classifications. The fully connected layer is followed by a nonlinear function, which is a softmax function that outputs real values ranging between 0 and 1 (Yamashita et al., 2018).

CNN's can be used to perform different computer vision tasks. However, this research will only focus on image classification.

### 2.2.3 Image Classification

Image classification is a computer vision task used to perform image recognition by taking an image as input and outputting a single class label or a distribution of probabilities of the image belonging to a particular class. Using the standard structure of CNN, one is equipped to build a simple or complex CNN model that performs image recognition. However, there are already existing CNN model architectures that have been shown to perform very well (Gu et al., 2017). Some of the common architectures include AlexNet, GoogleNet and Visual Geometry Group (VGG).

VGG secured the first and the second places in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) localisation and classification tasks respectively (Simonyan & Zisserman, 2015). The model also significantly outperformed the other models in a performance comparison study by (ul Hassan, 2018).

### 2.2.4 Transfer Learning

Transfer learning is a branch of machine learning that involves knowledge transfer from a source domain to a related target domain. It is unlikely to obtain a dataset large enough to train a whole CNN. Using transfer learning, a CNN is pre-trained on a large dataset and then used as a feature extractor (Chilamkurthy, 2020). In object classification, the initial layers of the CNN are fixed and only the last layers are trained.

In this research, the scarcity of poverty data is overcome by implementing two transfer learning steps. The first step involves transferring knowledge from ImageNet to extract low-level features like edges. The second transfer learning step involves adapting the model to satellite imagery input and learn features for predicting nighttime light intensities and poverty indices.

### 2.2.5 Poverty Measures

Poverty can be measured at either an individual or a household category. The latter category is preferred because individual income and expenditure is often pooled within the household (Govender et al., 2007). Monetary poverty is one of the main indicators of poverty and is usually measured in developing countries by consumption expenditure aggregates and a poverty line. The consumption expenditure aggregates are obtained by time-consuming and infrequent household surveys.

The Living Standards Measurement Study (LSMS) survey gathers information on a variety of aspects of household and individual well-being in order to measure household welfare and analyse the impact of various government policies on people's living situations in developing nations (Kinnon et al., 2001). The LSMS survey data is open to the public.

## 2.3 Related Work

Over the last few years, researchers have been combining different data sources with survey data to enhance the prediction of poverty using computational methods. This section reviews some of the methods used and the respective outcomes.

### 2.3.1 Nighttime light intensity as a measure of socio-economic development

(Xie et al., 2016) used nighttime light intensities as a proxy to predict and map poverty at a country or continental level. They implemented transfer learning by training a CNN model to predict nighttime light intensity from the daytime satellite images and simultaneously train another model that captures the effect of features on the satellite image to predict poverty. They obtained that daylight satellite images can be utilised to make relatively accurate spatial economic status predictions across

Nigeria, Uganda, Malawi, and Tanzania (Jean et al., 2016). Visualisation of the extracted image features suggests that the model learns to identify relevant socio-economic characteristics for the poverty estimation problem.

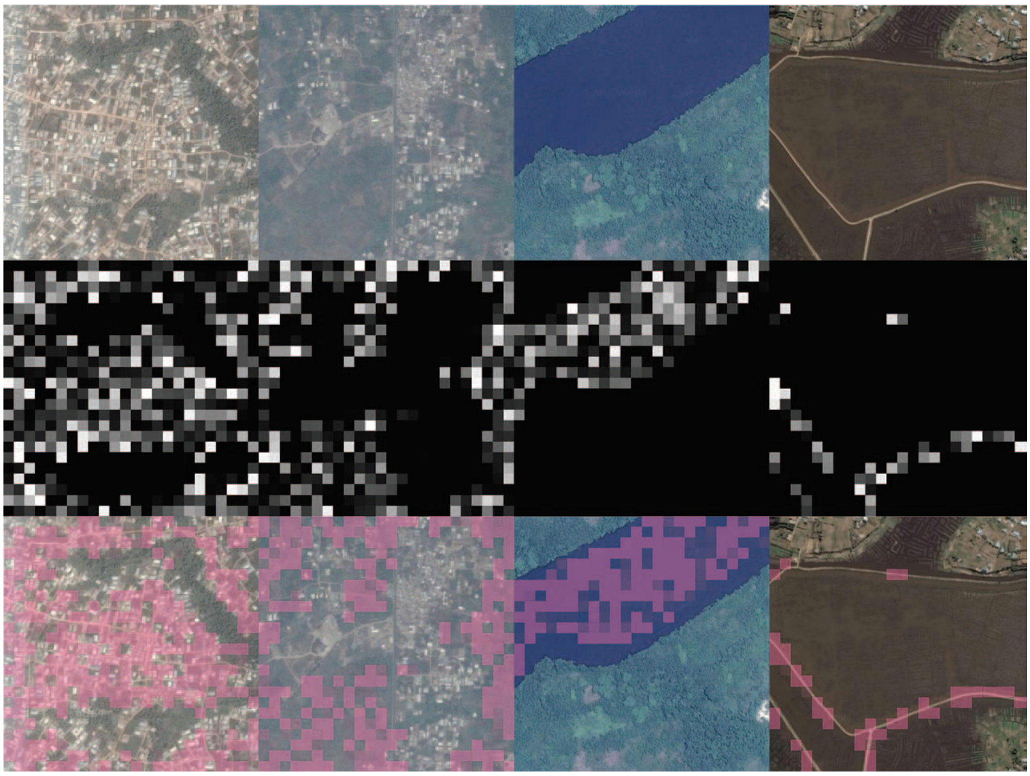


***Figure 2.3:*** *Convolutional filters highlighting the features extracted. Regions in pink are activated by the filter. Source : (Xie et al., 2016)*

(Xie et al., 2016)'s transfer learning approach of estimating poverty has outperformed models that are trained using passively collected data such as cellphone meta-data, because aerial images provide additional information that they leveraged on and used as a proxy for poverty. The results of research have ignited a field of potential ways of using satellite images to supplement survey data to perform studies that were data restricted due to poor available data.

(Head et al., 2017) explored the extent in which (Xie et al., 2016)'s approach to estimate poverty can be used to estimate a broader set of socio-economic indicators in different regions. (Head et al., 2017) demonstrated that (Xie et al., 2016) approach can be generalised to other countries and continents. However, they found that the performance of the model is sensitive to the model hyperparameters used. Furthermore, (Head et al., 2017) found that this approach does not generate a model that accurately estimates other socio-economic indicators such as access to water, electrification, education level and child weight-for-height-index, as it estimates asset-based wealth indexes such as the Gross Domestic Product (GDP). (Engstrom et al., 2017) further investigated the use of nighttime lights and high resolution satellite images to estimate poverty. They found that features from satellite images explains approximately 60% of the variation of poverty and a model built using night time lights explained 15% of the variation of poverty (Engstrom et al., 2017). They also found that built-up area and roof type have a strong correlation with welfare. These satellite object features were identified using deep learning-based CNN and classification of spectral and textural traits.

This research will follow a similar paradigm by using up-to-date data specifically nightlights, satellite imagery, survey data and machine learning tools.

### 2.3.2 Mapping poverty in Rwanda using mobile phone data

(Blumenstock et al., 2015) used anonymised data from Rwanda's largest mobile phone network to predict the subscribers' poverty status and further create a geographic distribution of wealth in Rwanda. They demonstrated that an individual's historic records of mobile phone usage can be utilised to infer the individual's socio-economic status and also accurately reconstruct the distribution of wealth in the country. (Blumenstock et al., 2015)'s approach involves combining feature engineering and feature selection, where they used the wealth index to measure poverty.

They trained an elastic net model using 5-fold Cross-Validation (CV) and found a strong correlation of $r = 0.916$ between district level wealth predicted by mobile phone data and average wealth of households from Demographic and Health Survey (DHS) data. They concluded that their approach can be used to approximate national wealth distribution in Rwanda. However, the reliance of anonymised phone usage data makes it an infeasible approach because some network providers have strict rules about sharing their subscribers' phone usage data.

### 2.3.3 Combining CNN and car imagery for prediction of demographic statistics

(Gebru et al., 2017) explored the use of survey data, presidential election voting data and collection of cars observed in an American neighbourhood using Google street view to estimate the neighbourhoods' demographics. Their findings suggest that demographic, socio-economic features and voting patterns can be extrapolated from the type of cars observed in neighbourhoods using Google street view (Gebru et al., 2017).

## 2.4 Research Gap

The main drawbacks of this poverty estimation approach, as demonstrated from the related studies, is the duration of labelling the images and training the CNN models. Manually labelling the images requires a lot of time. Other labelling methods can be that are more efficient and faster than manual labelling can be explored. There is transfer learning where, models trained learned previously are used to solve new problems faster or with better solutions (Pan & Yang, 2010). (Jean et al., 2016) used transfer learning but with older machine learning tools and different sources of data that are less accurate in comparison to current sources.

Supplementing the transfer learning approach with national household survey data provides a more frequent and automated method for updating poverty estimates that can be used for policymaking

## 2.5 Conceptual Model

The conceptual model for this research is as illustrated in Figure 2.4. The input data is satellite imagery that will be used to train the CNN models to classify nighttime light intensities. The extracted features are used in regression models for estimating the poverty rates.

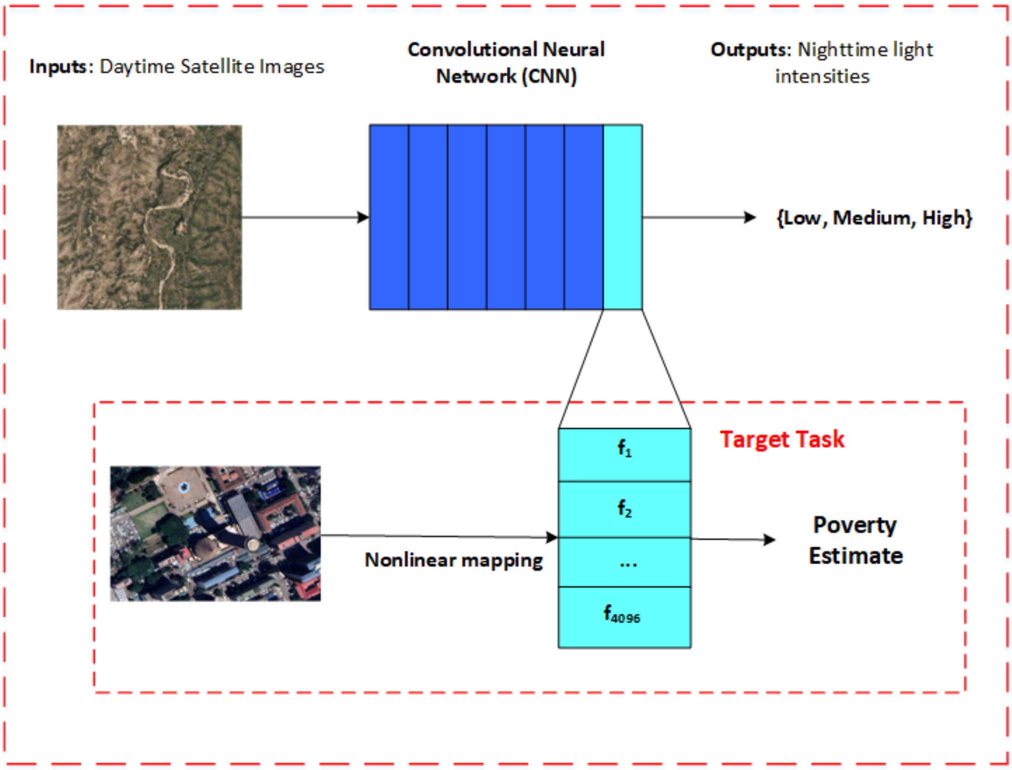



***Figure 2.4:*** *Conceptual Model*

# Chapter 3

# Research Methodology

## 3.1 Introduction

This chapter describes the activities undertaken in collecting the satellite imagery and survey data, the data description, techniques used in data preparation and analysis, design of the CNN and regression model and lastly evaluation of the models to ensure they achieve the research objectives.

The research utilises quantitative methodological approaches and follows the positivist paradigm.

## 3.2 Research Design

This research is guided by the CRoss Industry Standard Process for Data Mining (CRISP-DM). CRISP-DM is a model with six phases that describe the machine learning cycle (Vorhies, 2016). The phases include business understanding, data understanding, data preparation, modelling, evaluation and deployment as seen in Figure 3.1 below.

### 3.2.1 CRISP-DM Phases

1. The business understanding phase focuses on understanding the measures of poverty and transferring this knowledge into a machine learning problem.

2. Data understanding involves the data collection, description and verification.

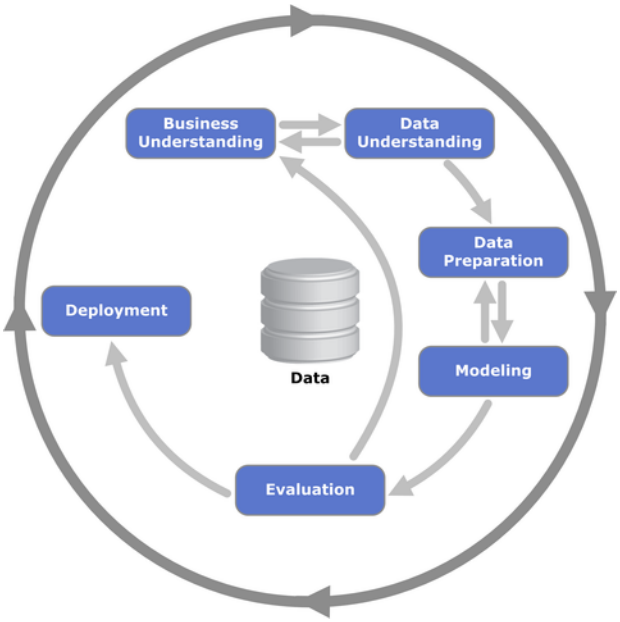



***Figure 3.1:*** *The CRISP-DM Cycle. Source : (Vorhies, 2016)*

LSMS surveys, daytime satellite imagery and nighttime light intensities are the data collected in this research.

3. Data preparation involves cleaning, transformation and integration of the different data sets. In this phase, the consumption expenditure per capita from the LSMS survey is mapped to the respective nightlight intensity using the geographic coordinates. Additionally, The nighttime light intensities are classified as either low, medium or high.

4. In the modelling phase, The machine learning is model selected and developed. The CNN and regression models used in this research are built in this phase. The CNN model is used in prediction of nighttime light intensity from daytime satellite imagery used as inputs for feature extraction. The regression model is used in poverty estimation using extracted features of daytime satellite imagery and actual consumption metrics from the survey data.

5. Evaluation entails a review of the performance of the models and if they meet the objectives. Cross-Validation (CV) is used for the model evaluation.

6. The last phase is deployment and consists of presenting the results in an understandable manner.

## 3.3 Data Description

The data sources for this research are daytime satellite imagery, nighttime light intensities and survey data from the Living Standards Measurement Study (LSMS). The data will be obtained in the same context and concurrently. This section describes the data sources in detail.

### 3.3.1 Survey Data

The research uses consumption expenditure as a socio-economic indicator, as measured in the LSMS survey. Annual household consumption expenditure is the amount of final consumption expenditure made by the household residents over a period of 12 months in order to meet their everyday needs. The LSMS surveys used in this research are: Ethiopia 2015-16, Malawi 2016-17 and Nigeria 2015-16. LSMS surveys use a two-stage sampling design, in which enumeration areas, referred to as "clusters" in this research are sampled randomly throughout a nation, with the probability of sampling proportionate to the population, and then households within each cluster are sampled randomly (Xie et al., 2016). To minimize error, only cluster coordinates with multiple households are selected when training and evaluating the model.

For each survey, the average household consumption expenditures at the cluster level are calculated, and the measurements converted to a common currency of the United States Dollar (USD) using purchasing power parity exchange rates for the respective survey years, allowing for direct comparison to the current World Bank global poverty line of $1.90 per capita per day.

To maintain the anonymity of survey respondents, the cluster locations are calculated using the average latitude and longitude coordinates. Due to the fact that the true location of each cluster is uncertain, daylight satellite imagery is sampled from a 10 km by 10 km square centered on the claimed cluster location and then averaged across the whole region. Additionally, each home is assigned a binary poverty classification based on survey consumption expenditure data. The consumption expenditure of households within each cluster determines the overall poverty designation for that cluster. The extracted features are then used to estimate the consumption expenditure. A summary of the LSMS dataset is seen in Table 3.1 below.

| Country | No. of Households | Clusters |
|---|---|---|
| Ethiopia | 4,954 | 523 |
| Malawi | 12,447 | 780 |
| Nigeria | 4,590 | 664 |
| **Total** | **21,991** | **1,967** |

***Table 3.1:*** *Summary of LSMS survey data*

### 3.3.2 Nighttime Lights Data

Observing the planet's nighttime lights provides a unique perspective of the global landscape. In contrast to daylight remote sensing, there are several sources of lighting at night. Moonlight is one of these sources, as is light emitted directly from sources such as buildings, and light reflected by the ground, also referred to as surface albedo (NASA, 2019).

The Earth Observation Group (EOG) began producing Nighttime lights in 1994, using the Operational Linescan System (OLS) on Defense Meteorological Satellite Program (DMSP) satellites (Elvidge et al., 2021). The launch of modern, advanced satellites such as the Joint Polar-orbiting Satellite System (JPSS) in 2013, the Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB) on JPSS satellites has resulted in a massive improvement in low light detection in comparison to DMSP. EOG is able to leverage this technical advancement to provide users with superior-quality worldwide Nighttime lights data (Elvidge et al., 2021).

Nighttime lights have proven to be a good proxy for economic development. As seen in Figure 3.2, the dark northern region is North Korea and the southern region with abundant nightlights is South Korea. This illustrates that nighttime lights are indicators of how developed an area is.

Since 2012, EOG has produced a different dataset of nighttime lights derived from the VIIRS DNB. VIIRS data are produced on a 15-arc-second global grid, which is double the resolution of the current DMSP output. The VIIRS nightlights used in the research are from the 2015 annual composite in the 75N/060W tile and the 00N/060W tile where the target countries are located.

***Figure 3.2:*** *The difference in nighttime light intensities of North Korea and South Korea. Source : (Balazh, 2016)*

The consumption predicted by the transfer learning approach is compared to that calculated from nightlights using cluster-level coordinates provided by LSMS. To estimate the intensity of nightlights in a specific place, we utilise the dataset for the year the survey was performed, extract all values for regions within a 10km $\times$ 10km square centred on the specified coordinates, and assign the cluster the mean value.

### 3.3.3 Daytime Satellite Imagery

As satellite imagery becomes widely available at the global scale, an abundance of information on landscape features that might be correlated to economic outcomes is being collected.(Xie et al., 2016).

The daytime satellite imagery were collected from Google Maps at zoom level 16 and size 400 by 400 pixels, which roughly corresponds to an area of $1km^2$.

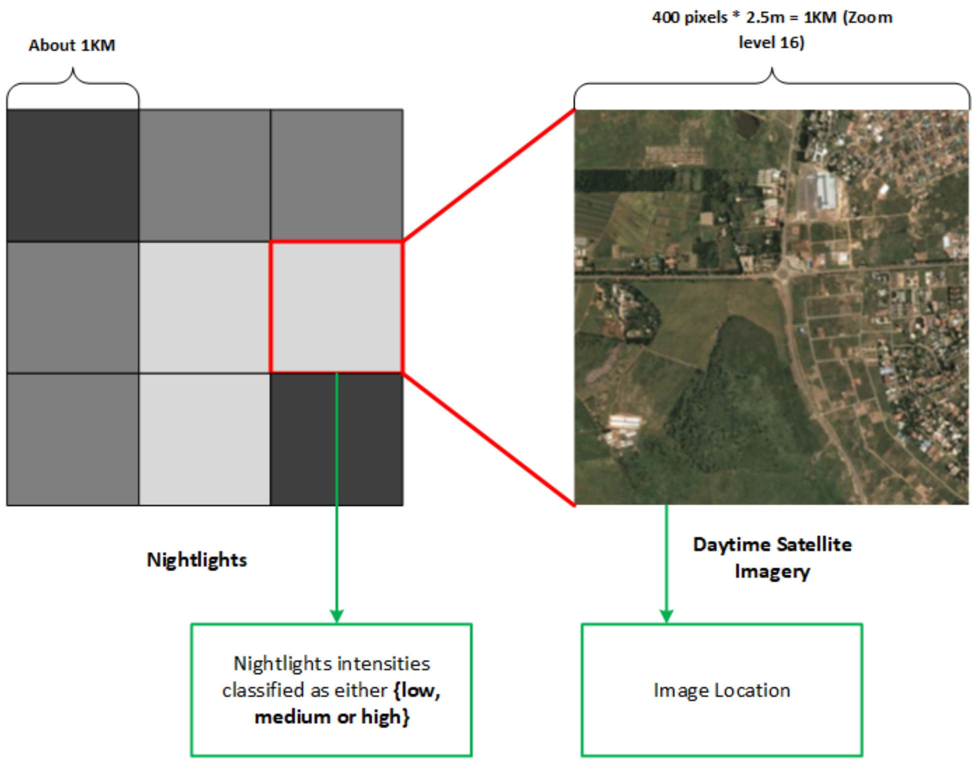



***Figure 3.3:*** *Daytime satellite imagery with nighttime light intensity as labels*

### 3.3.4 Target Countries

Ethiopia, Malawi and Nigeria were selected as the target countries for this research for the following reasons:

1. Availability of LSMS survey data: The LSMS initiative emerged from the need to enhance statistical data at the household level, which is necessary for developing nations to implement and evaluate social and economic policies (Kinnon et al., 2001).

2. Extending earlier research: Over the last two decades, poverty in developing countries has been studied largely and merits further research. For example, a study by (Jean et al., 2016) introduces a convolutional CNN model that learns to identify features such as buildings, roads, water bodies and farmlands. Using these features, the model performed closely to data collected in the field for poverty estimation in Uganda, Tanzania and Nigeria using the limited survey data and abundant nightlights data as a data-rich proxy.

## 3.4 Model Design and Development

### 3.4.1 Convolutional Neural Network Model

Deep learning approaches, particularly convolutional neural networks, have accelerated recent breakthroughs in computer vision, aided by massive datasets such as ImageNet, which includes millions of labeled training images (Russakovsky et al., 2015). The CNN model used in this research is highly nonlinear, has over 55 million parameters, and is sufficiently flexible to extract complex features from images, e.g., the presence or absence of a road. Since there is a scarcity of data on consumption expenditure in each country to be used as labelled training examples, a CNN model cannot be used to estimate these outcomes from satellite images.

To combat the data scarcity problem, transfer learning is used to train a fully-convolutional CNN model to predict nighttime light intensities from satellite imagery. By solving this related problem, the model learns how to extract features relevant for poverty prediction. In previous work (Xie et al., 2016), a multi-step approach outperforms simpler transfer learning methods that use imagery and nightlight information.

#### VGG

The VGG network is an eight layer CNN, see Figure 3.4, which has been originally designed and trained for image classification. It takes a fixed-size input image of 224 x 224 pixels. The network has been trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) data using gradient descent with momentum. Transfer learning is used to train the VGG model on the satellite images to predict the nighttime light intensities.

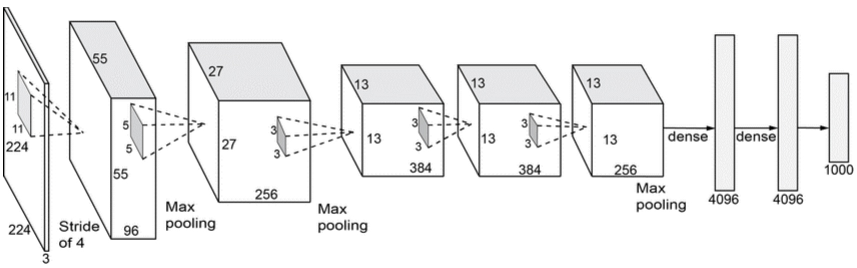



***Figure 3.4:*** *VGG Architecture*

In the initial step, an 8-layer CNN model (VGG) obtained from the PyTorch machine learning framework (PyTorch, 2019) and previously trained on ImageNet is fine-tuned to estimate nighttime light intensity given the corresponding daytime satellite images as inputs (Chatfield et al., 2014). This is a classification stage, with three classes of nighttime light intensity determined by fitting the relative frequencies of the nighttime light intensity values to a combination of three Gaussian distributions. The three categories of nighttime light intensity are low, medium, and high (Jean et al., 2016). The training set contained over 98,000 locations in the three target countries. The nightlight intensity values are integer values ranging from 0-63. This interval is divided into a low class with values 0-3, a medium class with values ranging between 3-34 range, and a high class with values between 35-63.

The model is trained with mini-batch gradient descent with momentum update. After training the CNN model to predict nighttime light intensity, the model is used as a feature extractor for daytime satellite images by removing the last layer of the CNN model, which is the nighttime light classification layer.

For each household cluster, the input images cover a 10 km by 10 km area centred around the cluster location. Therefore, for each country, the image features per cluster are aggregated and then used as input in the regression models for estimating survey-based measures of consumption expenditure.

### 3.4.2 Regression Model

Regression analysis is a collection of statistical techniques used to identify connections between interrelated variables. (Golberg & Cho, 2004). Regression includes several variations, such as linear, multiple linear, and nonlinear. Linear regression has a couple of weaknesses such as overfitting and sensitivity to outliers (Frost, 2020). To overcome these issues, statisticians developed a number of advanced variations, like Ridge regression, Least Absolute Shrinkage and Selection Operator (LASSO) regression and Partial Least Squares regression (Frost, 2020).

Ridge Regression is a technique for analysing multiple regression data containing near-linear relationships among the independent variables (NCSS, 2021). It enforces squared penalties on the size of the linear coefficients. Ridge regression will be used in this research for prediction of consumption expenditures using the extracted cluster features.

## 3.5 Model Evaluation

### 3.5.1 Cross-Validation

Cross-Validation (CV) is a statistical method used to estimate the performance of machine learning models (Brownlee, 2018). The procedure is often referred to as $k$-fold cross-validation, with $k$ being the number of sets the data is split into.

The ridge regression models are evaluated in 5-fold cross-validation for each country and all countries combined. Since the dimension of the image features used is large ($d = 4096$), regularisation helps to prevent overfitting to the relatively small training sets. The model $R^2$ reported is the average test $R^2$ across the cross-validation folds.

## 3.6 Development Environment

### 3.6.1 Hardware and Software Specifications

The models will be implemented on a virtual machine on Google Cloud Platform (GCP) with specifications in Table 3.2 below.

| Specification | Description |
|---|---|
| Operating System | Debian GNU/linux |
| GPU | 1 NVIDIA Tesla K80 GPU |
| CPU | 4 vCPU |
| Memory | 26GB |
| Region | us-west1 (Oregon) |
| Zone | us-west1-b |
| Machine Learning Tools | PyTorch, Jupyter Notebooks |

***Table 3.2:*** *Development Environment*

## 3.7 Limitations of Methodology

The number of variables possible in the image classification phase is broad but in this research only consumption expenditure was considered. The small number of model runs was selected due to limited computing resources, because a high number of model runs and regression K-folds results in increased computing time and effort.

# Chapter 4

# Results and Discussion

## 4.1 Introduction

This chapter discusses the findings of the research. The evaluation techniques described in Chapter 3 are used to analyse the performance of the CNN and regression models.

## 4.2 Results

### 4.2.1 Nighttime Lights Prediction

The CNN model attained an accuracy of 75.29% on the training data and 77.97% on the validation data over 81,000 iterations as seen in Figure 4.1 below.

The pre-trained VGG model was downloaded from the PyTorch Model Zoo (PyTorch, 2019). Mini-batch gradient descent with momentum was used for optimisation and Cross-Entropy (CE) as the loss function. Normalisation and data augmentation were implemented in the training set. Random cropping and random mirroring were used for the data augmentation. Regularisation was achieved using a dropout of 50% on the dense layers. The learning rate was initialised at $10^{-4}$. The model was trained in roughly 6 hours. The last layer of the trained model was used to extract 4,096 features used to predict the average per capita consumption of clusters, taken from the LSMS survey.

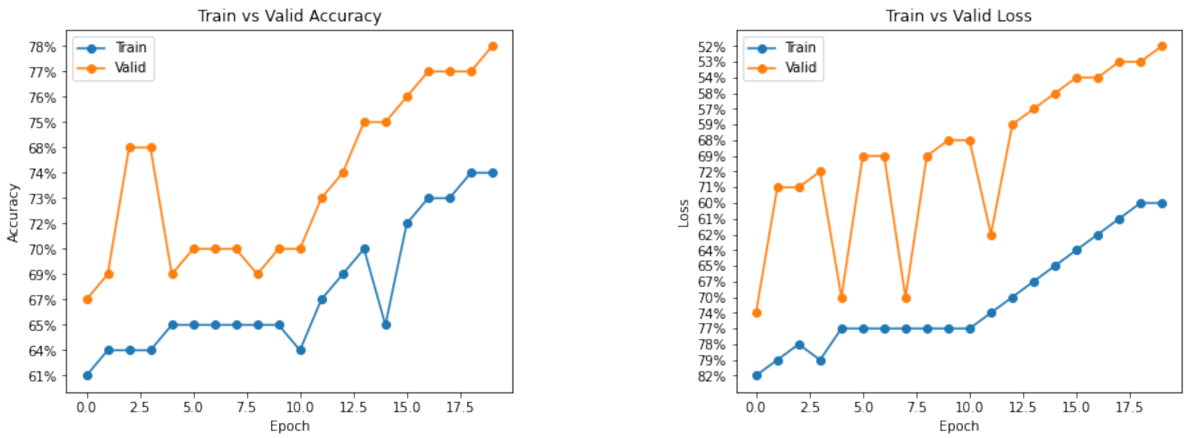



***Figure 4.1:*** *Plots of accuracy and loss on the training and validation datasets over 20 epochs.*

### 4.2.2 Poverty Estimation

The relationships between predicted and actual consumption for the countries is as illustrated in Figure 4.2 below. Each data point in the scatter-plots represents a cluster in the LSMS survey. The x-axis represents the cluster's actual consumption expenditure while the y-axis represents the cluster's predicted consumption expenditure. The model explains 18 to 43% of the variation in average household expenditures in each country.

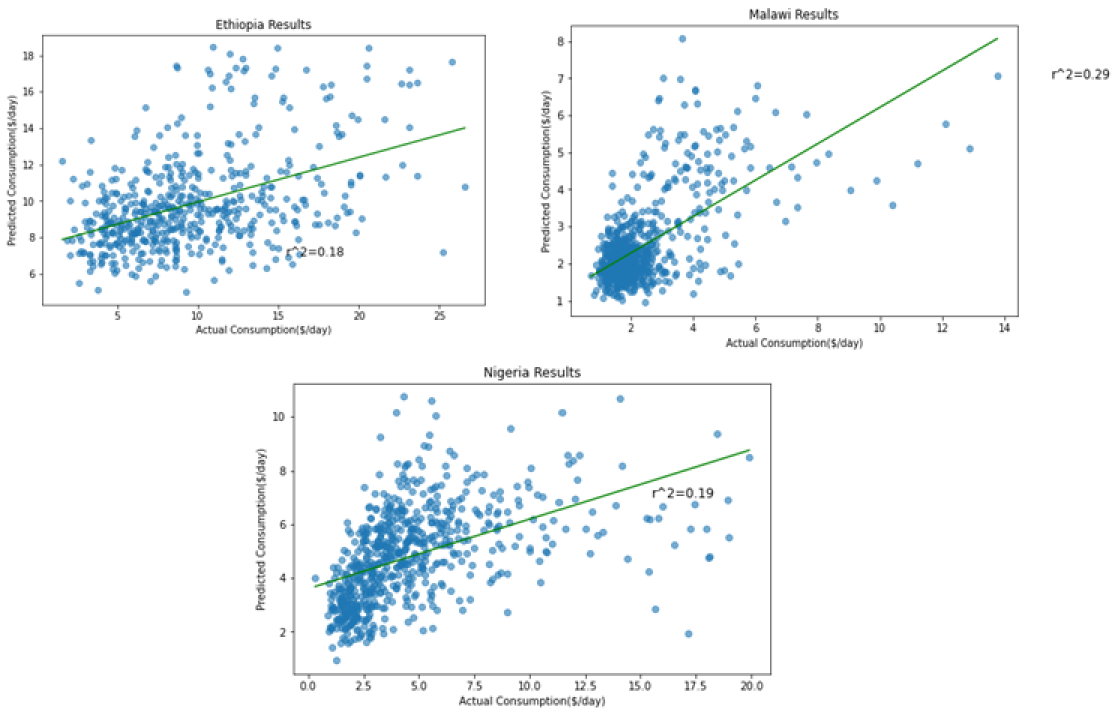



***Figure 4.2:*** *Relationship between predicted and actual consumption per country.*

Randomised 5-fold Cross-Validation (CV) on all countries, randomised 5-fold CV per country, spatial 5-fold CV and cross-country CV were used to evaluate the model. Results are as seen in Figure 4.3 and Tables 4.1 and 4.2 below:

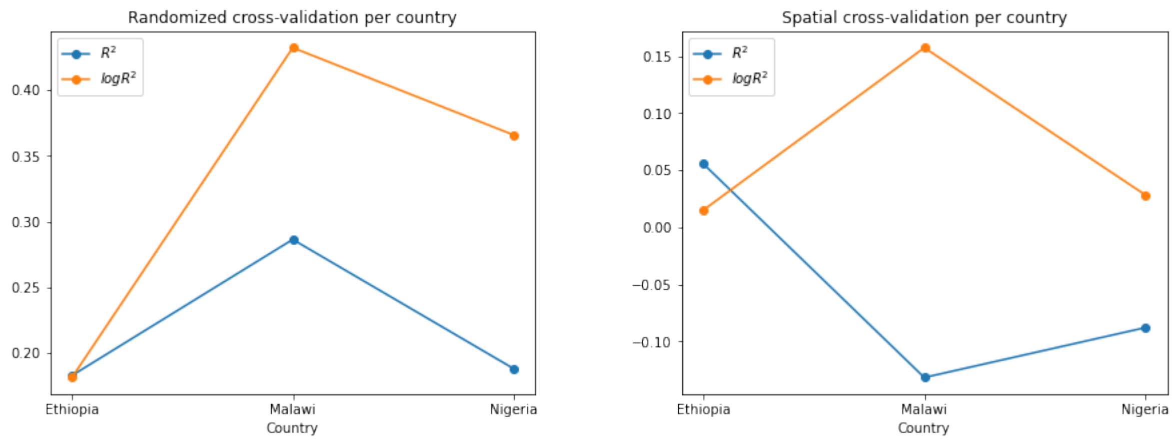



***Figure 4.3:*** *Randomised and Spatial Cross-Validation per country.*

| Country | $R^2$ | $logR^2$ |
|---|---|---|
| All Countries | -1.146 | -3.4373 |

***Table 4.1:*** *Cross-country Cross-Validation per Country*

| Country | $R^2$ | $logR^2$ |
|---|---|---|
| All Countries | 0.3325 | 0.4794 |

***Table 4.2:*** *Randomised cross-validation for all countries*

### 4.2.3 Visualisation of Extracted Features

Due to the large number of daytime satellite imagery, it is difficult to manually review each image and the corresponding activation map so as to make generalisations about the model's focus areas. Visualisation of a random sample demonstrates that roads and the edges of water bodies tend to be identified as seen in Figure 4.4 below. These findings suggest that transfer learning could be used to estimate poverty using data-rich sources that are publicly available and inexpensive.

### 4.2.4 Comparison to Using Survey Data Only

To evaluate the model's accuracy, features from the LSMS survey that an image can identify were extracted and used to predict consumption. This serves as the benchmark for image-based models.

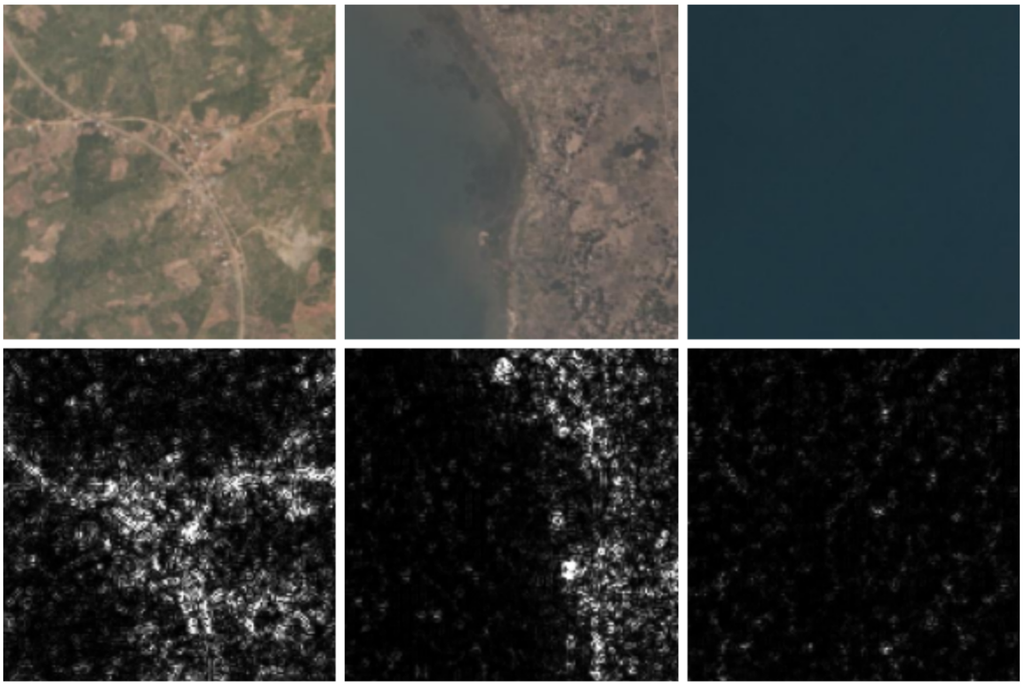


***Figure 4.4:*** *Activation maps highlighting extracted features.*

Features from the Malawi LSMS survey data that an image could recognise were aggregated and used to predict consumption. The $R^2$ results are seen in Table 4.3 below:

| **Country** | $R^2$ | $logR^2$ |
|---|---|---|
| Malawi | 0.088 | 0.4664 |

***Table 4.3:*** *Poverty prediction model built using only survey data*

The performance of the transfer model based on remotely sensed data is comparable to that of the survey model based on data gathered in the field at a high cost.

## 4.3 Discussion

The results demonstrate that the transfer learning technique to estimating poverty indices using daytime and nighttime satellite imagery might be beneficial in developing countries where poverty data is scarce. In addition to being inexpensive, scalable and accurate, this approach has huge potential as a complement or alternative to traditional data collection methods. The model performs well despite instances of imprecise periods between the satellite imagery and cluster locations. In each of these dimensions, more accurate data is likely to increase the model's performance.

However, there are potential limitations in using this approach. For example, looking at the extracted features from the satellite images in Figure 4.4, paved roads, built-up areas and so forth are all features that one could associate with affluence. Other poverty indicators such as child mortality or level of education may not be identified from satellite imagery. If locations with high and low rates of child mortality seem identical, it may be difficult to estimate the prevalence of child mortality from satellite imagery. Additionally, poverty is a condition that applies to a group or an individual, whereas satellite images capture the characteristics of a location.

Finally, the transfer learning technique could be further evaluated by the incorporation of higher resolution time-series data of daytime imagery to track changes in economic outcomes or as a complement to other approaches such as mobile phone metadata (Blumenstock et al., 2015).

# Chapter 5

# Conclusions and Recommendations

## 5.1 Introduction

This chapter provides an assessment of the value of this study, conclusions inferred from the study results, fundamental limitations of the study and recommendations for future work.

## 5.2 Conclusion

The main objective of this research was to implement a CNN model for predicting poverty using satellite imagery. This was achieved by implementing a transfer learning method by first training the CNN model to predict nighttime light intensity from daytime satellite imagery. The trained model was then used to extract relevant features that were used in a regression model with household consumption expenditure as the poverty labels to predict poverty. The model was able to explain 18% to 43% variation in household expenditure across the three countries.

This approach utilizes abundant sources of publicly available data that provide a solution to the lack of poverty data in developing countries. The method can be used in estimating poverty data in the years when national surveys are not conducted and in real-time monitoring of poverty for effective policy implementations.

## 5.3 Limitations

There are a couple of reasons why it might be difficult to use satellite imagery to measure poverty accurately. These include:

1. The satellite imagery must include relevant data on the poverty indicators. Poverty indicators such as child mortality can not be identified from satellite imagery.

2. The study was based on an assessment of the model's generalizability to other African countries. However, developing countries in other continents could provide different results.

## 5.4 Recommendations

This research focused explicitly on using the transfer learning approach to extracting information from satellite images. Other approaches incorporating feature engineering by manually labelling the images are resource-intensive but will drastically improve the performance of the models. Incorporation of other sources of data, such as time-series data, will improve the accuracy of the data. Additionally, observing temporal trends enables the model to enhance its present forecasts.

# References


Balazh, A. (2016). Planet earth with detailed exaggerated relief at night lit by the lights of cities. https://www.shutterstock.com/g/antartis

Bengio, Y., & Glorot, X. (2010). Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, *9*, 249–256. https://doi.org/http://proceedings.mlr.press/v9/glorot10a.html

Blumenstock, J., Cadamuro, G., & On, R. (2015). Predicting poverty and wealth from mobile phone metadata. *Science*, *350*(6264), 1073–1076. https://doi.org/10.1126/science.aac4420

Bouvrie, J. (2006). Notes on convolutional neural networks. https://doi.org/http://cogprints.org/5869/

Brownlee, J. (2018). *A gentle introduction to k-fold cross-validation.* https://machinelearningmastery.com/k-fold-cross-validation/

Brownlee, J. (2020). *What is deep learning?* https://machinelearningmastery.com/what-is-deep-learning

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets.

Chilamkurthy, S. (2020). *Transfer learning for computer vision.* https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

Elvidge, C. D., Zhizhin, M., Ghosh, T., Hsu, F.-C., & Taneja, J. (2021). Annual time series of global viirs nighttime lights derived from monthly averages: 2012 to 2019. *Remote Sensing*, *13*(5). https://doi.org/10.3390/rs13050922

Engstrom, R., Hersh, J., & Newhouse, D. (2017). Poverty from space : Using high-resolution satellite imagery for estimating economic well-being. *Policy Re-*

*search Working Paper*, (8284). https://doi.org/http://hdl.handle.net/10986/29075

Frost, J. (2020). *Choosing the correct type of regression analysis*. https://statisticsbyjim.com/regression/choosing-regression-analysis/

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E. L., & Fei-Fei, L. (2017). Using deep learning and google street view to estimate the demographic makeup of the us.

Golberg, M., & Cho, H. (2004). *Introduction to regression analysis*. WIT Press. https://books.google.co.ke/books?id=zScAAjY3BDcC

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Govender, P., Kambaran, N., Patchett, N., Ruddle, A., Torr, G., & Zyl, N. V. (2007). Poverty and inequality in south africa and the world authors. *ASSA's Social Security Committee*.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G., Cai, J., & Chen, T. (2017). Recent advances in convolutional neural networks.

Head, A., Manguin, M., Tran, N., & Blumenstock, J. E. (2017). Can human development be measured with satellite imagery? *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*. https://doi.org/10.1145/3136560.3136576

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *354*(6301), 790–794. https://doi.org/https://doi.org/10.1126/science.aaf7894

Jütting, J., & Donnell, I. M. (2017). Overview: What will it take for data to enable development? *Development Co-operation Report 2017: Data for Development*. https://doi.org/https://doi.org/10.1787/dcr-2017-6-en

Kinnon, S., Steele, D., & Temesgen, T. (2001). Living standards measurement study surveys.

Murthy, K., Shearn, M., Smiley, B. D., Chau, A. H., Levine, J., & Robinson, D. (2014). Skysat-1: Very high-resolution imagery from a small satellite. *Proceed-*

*ings of SPIE - The International Society for Optical Engineering*, 92411E–92411E. https://doi.org/http://dx.doi.org/10.1117/12.2074163

NASA. (2019). *Nighttime lights*. https://earthdata.nasa.gov/learn/backgrounders/nighttime-lights

NCSS. (2021). *Regression analysis in ncss*. https://www.ncss.com/software/ncss/regression-analysis-in-ncss/

Nielsen, M. (2019). *Neural networks and deep learning*. http://neuralnetworksanddeeplearning.com

Okalow, S. (2021). *What is poverty? it's not as simple as you think*. https://www.worldvision.ca/stories/child-sponsorship/what-is-poverty

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

PyTorch. (2019). *Pytorch documentation*. https://pytorch.org/docs/stable/index.html

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2014). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 1–42. https://doi.org/https://doi.org/10.1007/s11263-015-0816-y

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition.

ul Hassan, M. (2018). *Vgg16 – convolutional network for classification and detection*. https://neurohive.io/en/popular-networks/vgg16/

Vorhies, W. (2016). *Crisp-dm – a standard methodology to ensure a good outcome*. https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome

World Bank. (2015). *Povcalnet online poverty analysis tool,* http://iresearch.worldbank.org/povcalnet/

Xie, M., Jean, N., Burke, M., Lobell, D., & Ermon, S. (2016). Transfer learning from deep features for remote sensing and poverty mapping. *30th AAAI Conference on Arti cial Intelligence*. https://doi.org/https://arxiv.org/abs/1510.00098v2

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Imaging*, 611–629. https://doi.org/https://doi.org/10.1007/s13244-018-0639-9

Zeiler, M. D., & Fergus, R. (2013). Visualizing and understanding convolutional networks.

# Appendix A

# Code Snippets

## A.1 Processing Survey Data

Household LSMS survey data : https://microdata.worldbank.org/index.php/catalog/lsms

Nightlights data : https://ngdc.noaa.gov/eog/viirs/download_dnb_composites.html

Functions to output a dataframe with the following columns:

country, cluster_lat, cluster_lon, cons_pc

Each row should represent one cluster by combining the household data

’’’

```
import pandas as pd
import numpy as np
import os
import geoio

pd.set_option(’mode.chained_assignment’, None)

BASE_DIR = ’..’
NIGHTLIGHTS_DIRS = [os.path.join(BASE_DIR, ’data/nightlights/viirs_2015_00N060W.tif’),
                    os.path.join(BASE_DIR, ’data/nightlights/viirs_2015_75N060W.tif’)]

COUNTRIES_DIR = os.path.join(BASE_DIR, ’data’, ’countries’)
```

```
def process_malawi():
    lsms_dir = os.path.join(COUNTRIES_DIR, 'malawi_2016', 'LSMS')
    consumption_file = 'IHS4 Consumption Aggregate.csv'
    consumption_ph_col = 'rexpagg' # per household
    hhsize_col = 'hhsize' # people in household

    geovariables_file = 'HouseholdGeovariables_csv/HouseholdGeovariablesIHS4.csv'
    lat_col = 'lat_modified'
    lon_col = 'lon_modified'

    # purchasing power parity for malawi in 2016
    ppp = 215.182

    for file in [consumption_file, geovariables_file]:
        assert os.path.isfile(os.path.join(lsms_dir, file)), print(f'Could not find {file}')

    df = pd.read_csv(os.path.join(lsms_dir, consumption_file))
    df['cons_ph'] = df[consumption_ph_col]
    df['pph'] = df[hhsize_col]
    df['cons_ph'] = df['cons_ph'] / ppp / 365
    df = df[['case_id', 'cons_ph', 'pph']]

    df_geo = pd.read_csv(os.path.join(lsms_dir, geovariables_file))
    df_cords = df_geo[['case_id', 'HHID', lat_col, lon_col]]
    df_cords.rename(columns={lat_col: 'cluster_lat', lon_col: 'cluster_lon'}, inplace=True)
    df_combined = pd.merge(df, df_cords, on='case_id')
    df_combined.drop(['case_id', 'HHID'], axis=1, inplace=True)
    df_combined.dropna(inplace=True) # can't use na values

    df_clusters = df_combined.groupby(['cluster_lat', 'cluster_lon']).sum().reset_index()
    df_clusters['cons_pc'] = df_clusters['cons_ph'] / df_clusters['pph'] #
```

```
    df_clusters['country'] = 'mw'
    return df_clusters[['country', 'cluster_lat', 'cluster_lon', 'cons_pc']]
```

## A.2 Downloading of Daytime Satellite Images

```
def download_images(df):

    imd = GoogleDownloader()
    num_retries = 20
    wait_time = 0.1 # seconds
    zoom = 16

    # drops what is already downloaded
    already_downloaded = os.listdir(os.path.join(COUNTRIES_DIR, 'malawi_2016', 'images')) + \
                         os.listdir(os.path.join(COUNTRIES_DIR, 'ethiopia_2015', 'images')) + \
                         os.listdir(os.path.join(COUNTRIES_DIR, 'nigeria_2015', 'images'))
    already_downloaded = list(set(already_downloaded).intersection(set(df['image_name'])))
    print('Already downloaded ' + str(len(already_downloaded)))
    df = df.set_index('image_name').drop(already_downloaded).reset_index()
    print('Need to download ' + str(len(df)))
    # use three years of images to find one that matches search critera
    min_year = 2014
    min_month = 1
    max_year = 2016
    max_month = 12
    for _, r in tqdm(df.iterrows(), total=df.shape[0]):
        lat = r.image_lat
        lon = r.image_lon
        name = r.image_name
        country_dir = None
        if r.country == 'mw':
            country_dir = 'malawi_2016'
```

```
elif r.country == 'eth':
    country_dir = 'ethiopia_2015'
elif r.country == 'ng':
    country_dir = 'nigeria_2015'
else:
    print(f"unrecognized country: {r.country}")
    raise ValueError()
image_save_path = os.path.join(COUNTRIES_DIR, country_dir, 'images', r.image_name)
try:
    im = imd.download(lat, lon, zoom)
    if (type(im) == str and im == 'RETRY') or im is None:
        resolved = False
        for _ in range(num_retries):
            time.sleep(wait_time)
            im = imd.download(lat, lon, zoom)
            if (type(im) == str and im == 'RETRY') or im is None:
                continue
            else:
                plt.imsave(image_save_path, im)
                resolved = True
                break
        if not resolved:
            print(f'Could not download {lat}, {lon} despite several retries and waiting')
            continue
        else:
            pass
    else:
        # no issues, save according to naming convention
        plt.imsave(image_save_path, im)

except Exception as e:
    logging.error(f"Error—could not download {lat}, {lon}", exc_info=True)
```

```
            continue
```

## A.3 CNN Model Training

```
def initialize_model(model_name, num_classes, feature_extract, use_pretrained=True):
    # Initialize these variables which will be set in this if statement. Each of these
    # variables is model specific.
    model_ft = models.vgg11_bn(pretrained=use_pretrained)
    set_parameter_requires_grad(model_ft, feature_extract)
    num_ftrs = model_ft.classifier[6].in_features
    model_ft.classifier[6] = nn.Linear(num_ftrs,num_classes)
    input_size = 224
    return model_ft, input_size

def set_parameter_requires_grad(model, feature_extracting):
    if feature_extracting:
        for param in model.parameters():
            param.requires_grad = False

def train_model(model, dataloaders, criterion, optimizer, num_epochs=25):
    since = time.time()

    val_acc_history = []

    best_model_wts = copy.deepcopy(model.state_dict())
    best_acc = 0.0

    for epoch in range(num_epochs):
        print('Epoch {}/{}'.format(epoch, num_epochs - 1))
        print('-' * 10)
        if epoch > 10:
            # fine tune whole model
```

```
        for param in model_ft.parameters():
            param.requires_grad = True
        optimizer = optim.SGD(model_ft.parameters(), lr=1e-4, momentum=0.9)

    # Each epoch has a training and validation phase
    for phase in ['train', 'valid']:
        if phase == 'train':
            model.train() # Set model to training mode
        else:
            model.eval() # Set model to evaluate mode

        running_loss = 0.0
        running_corrects = 0

        # Iterate over data.
        for inputs, labels in tqdm(dataloaders[phase]):
            inputs = inputs.to(device)
            labels = labels.to(device)

            # zero the parameter gradients
            optimizer.zero_grad()

            # forward
            # track history if only in train
            with torch.set_grad_enabled(phase == 'train'):
                outputs = model(inputs)
                loss = criterion(outputs, labels)

                _, preds = torch.max(outputs, 1)

                # backward + optimize only if in training phase
                if phase == 'train':
```

```
                loss.backward()
                optimizer.step()

            # statistics
            running_loss += loss.item() * inputs.size(0)
            running_corrects += torch.sum(preds == labels.data)

        epoch_loss = running_loss / len(dataloaders[phase].dataset)
        epoch_acc = running_corrects.double() / len(dataloaders[phase].dataset)

        print('{} Loss: {:.4f} Acc: {:.4f}'.format(phase, epoch_loss, epoch_acc))

        # deep copy the model
        if phase == 'valid' and epoch_acc > best_acc:
            best_acc = epoch_acc
            best_model_wts = copy.deepcopy(model.state_dict())
        if phase == 'valid':
            val_acc_history.append(epoch_acc)

    print()

time_elapsed = time.time() - since
print('Training complete in {:.0f}m {:.0f}s'.format(time_elapsed // 60, time_elapsed % 60))
print('Best val Acc: {:4f}'.format(best_acc))

# load best model weights
model.load_state_dict(best_model_wts)
return model, val_acc_history
```

## A.4 Poverty Prediction

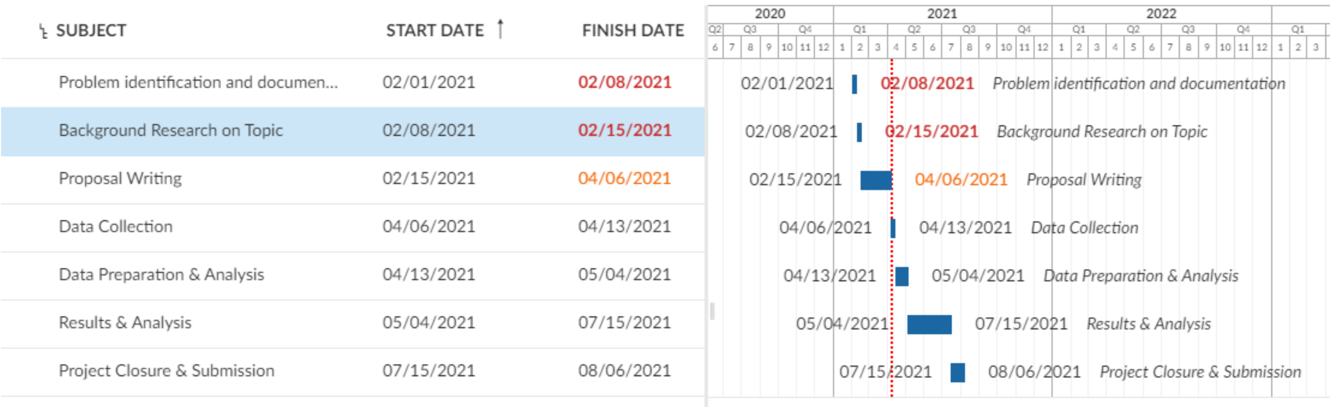
```
def plot_predictions(y, yhat, r2, country, max_y=None):
```

```
if max_y is not None:
    yhat = yhat[y < max_y]
    y = y[y < max_y]
fig = plt.figure(figsize=(8,5))
plt.scatter(y, yhat, alpha=0.6)
plt.plot(np.unique(y), np.poly1d(np.polyfit(y, yhat, 1))(np.unique(y)), color='g')
plt.text(15.5, 7, f'r^2={round(r2, 2)}', size=12)
plt.xlabel('Actual Consumption($/day)')
plt.ylabel('Predicted Consumption($/day)')
plt.title(f'{country} Results')
return fig
```

# Appendix B

# Project Schedule

## B.1 Gantt Chart



| SUBJECT | START DATE | FINISH DATE |
|---|---|---|
| Problem identification and documen... | 02/01/2021 | 02/08/2021 |
| Background Research on Topic | 02/08/2021 | 02/15/2021 |
| Proposal Writing | 02/15/2021 | 04/06/2021 |
| Data Collection | 04/06/2021 | 04/13/2021 |
| Data Preparation & Analysis | 04/13/2021 | 05/04/2021 |
| Results & Analysis | 05/04/2021 | 07/15/2021 |
| Project Closure & Submission | 07/15/2021 | 08/06/2021 |



***Figure B.1:*** *Project Work Plan.*