IMAGE CAPTION GENERATION USING VGG16 AND LSTM

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

JANNATHA MANISH	**20UK1A05C0**
GULLAPALLY GAYATHRI	**20UK1A05B7**
BASKE ARAVIND	**20UK1A0583**
JAKKU SHIVA	**20UK1A0562**

Under the esteemed guidance of

MS. V. Madhavi

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD
BOLLIKUNTA, WARANGAL – 506005

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH
HYDERABAD BOLLIKUNTA,
WARANGAL – 506005

<u>CERTIFICATE OF COMPLETION UG PROJECT PHASE-1</u>

This is to certify that the UG Project Phase-1 Report entitled **“IMAGE CAPTION GENERATION USING VGG16 AND LSTM”** is being submitted by **JANNATHA MANISH(20UK1A05C0), GAYATHRI GULLAPALLY(20UK1A05B7), BASKE ARAVIND(20UK1A0583), JAKKU SHIVA(20UK1A0562)** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2023-24, is a record of work carried out by them under the guidance and supervision.

Project guide	**Head of the department**
Ms.V.Madhavi	**Dr. R. Naveen Kumar**
(Asst professor)	(professor)

External

ACKNOWLEDGEMENT

I wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

I extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

I express heartfelt thanks to the guide, **Ms.V.Madhavi**, Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, I express my sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This project focuses on image caption generation using a combination of VGG16, a convolutional neural network for image feature extraction, and LSTM (Long Short-Term Memory), a recurrent neural network for sequence generation. The VGG16 model is employed to extract relevant features from input images, and these features are then fed into an LSTM network to generate descriptive captions. The synergistic integration of these two architectures aims to enhance the accuracy and richness of image captions, leveraging both visual context and sequential dependencies for more contextually relevant and coherent descriptions.

VGG16 extracts intricate visual features from input images, providing a comprehensive representation. Following this, LSTM, with its proficiency in capturing sequential dependencies, generates coherent and contextually relevant captions. The integration of these architectures bridges the gap between visual context and linguistic expression, enhancing the overall captioning performance. Through rigorous experimentation and evaluation, this research demonstrates the efficacy of the VGG16-LSTM model in producing accurate and meaningful image captions. The proposed methodology not only contributes to advancing image understanding but also highlights the potential of synergistic CNN-RNN models in capturing the intricate interplay between visual and linguistic elements for enhanced image description generation.

1. INTRODUCTION

1.1 OVERVIEW

An image caption description is a written caption that explains the important details of a picture. It involves generating a human readable textual description given an image, such as a photograph. For a human, it is a very simple task, but for a computer it is extremely difficult since it requires both understanding the content of an image and how to translate this understanding into natural language.

Recently, deep learning methods have displaced classical methods and are achieving state-of-the-art results for the problem of automatically generating descriptions, called "captions," for images.

1.2 PURPOSE

In this project, we will see how deep neural network models can be used to automatically generate descriptions for images, such as photographs.. In this project, we use CNN and LSTM to generate the caption of the image. As the deep learning techniques are growing, huge datasets and computer power are helpful to build models that can generate captions for an image. This is what we are going to implement in this Python based project where we will use deep learning techniques like CNN and RNN.

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

Image caption generation using VGG16 and LSTM faces challenges such as limited context understanding due to the spatial focus of VGG16, fixed image size constraints causing potential information loss, and the risk of overfitting with limited datasets. Ambiguity in complex scenes, biases in training data, computational complexity, and possible semantic drift in LSTM pose additional hurdles. Moreover, the lack of interactivity in one-shot generation limits adaptability.

2.2 PROPOSED SOLUTION

To address challenges in VGG16 and LSTM-based image captioning, a proposed solution involves augmenting VGG16 with attention mechanisms to enhance context understanding and mitigate fixed image size constraints. Regularization techniques and diverse datasets can mitigate overfitting and biases. Advanced architectures, like transformer-based models, can replace or complement LSTM to capture long-range dependencies more effectively, reducing potential semantic drift. Additionally, incorporating user feedback loops enables interactivity for refining generated captions. These enhancements collectively aim to create a more robust and adaptive image captioning system, capable of handling diverse visual content with improved precision, generalization, and interactive capabilities.

3.THEORETICAL ANALYSIS

3.1 TECHNICAL ARCHITECTURE

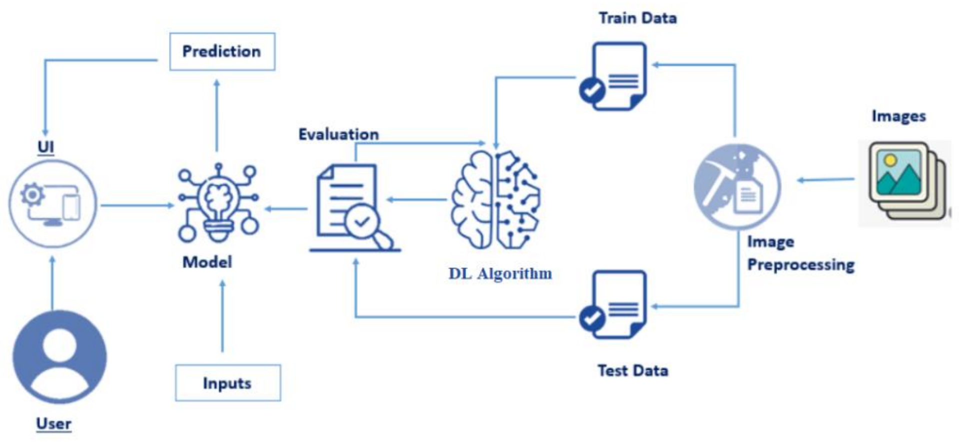


Figure 1: Architecture

3.2HARDWARE/SOFTWARE DESIGNING

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and user throughout the software development process.

The SRS may be one of a contract's deliverable data item descriptions or have other forms of organizationally- mandated content.

3.2.1 User Requirements

1.Good Practice:

For many projects, the total set of user requirements can be ambitious, making it difficult or even impossible to deliver a solution that meets all the requirements, in a way, that is robust, cost-effective, maintainable and can be rolled out quickly to a large user b important for global projects with a large user base. It is important to match the user requirements specification against the available technology and solutions that can be implemented in a timely, robust and practical way. This may result in an agreement that some of the requirements, say 20%, will not be delivered.

Such a compromise will make sure the remaining 80% can be delivered quickly. This compromiseis important for global projects with a large user base. On such projects, the speed and ease of implementation is an important consideration in the overall solution. To be successful at requirements gathering and to give your project an increased likelihood of success, follow these rules:

1.Don’t assume you know what the customer wants, ask!

2 .Involve the users from the start.

3.Define and agree on the scope of the project.

4.Ensure requirements are specific, realistic and measurable.

5.Get clarity if there is any doubt.

6.Create a clear, concise and thorough requirements document and share it with the

7.Confirm your understanding of the requirements with the customer by playing them back.

8.Avoid talking technology or solutions until the requirements are fully understood.

9.Get the requirements agreed with the stakeholders before the project starts.

10.Create a prototype, if necessary, to confirm or refine the customers’ requirements.

11.Cross-check the software design against the requirements and review regularly.

2.Common Mistakes:

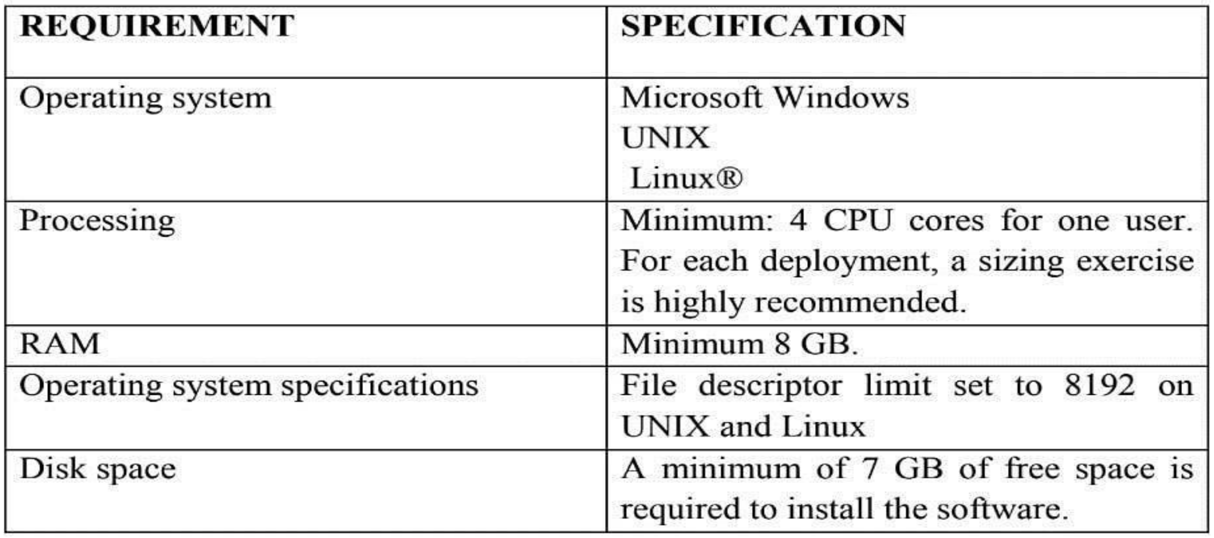
Basing a solution on complex or new technology and then discovering that it cannot easily be rolled out to the 'real world.'

- Not prioritizing the User Requirements into 'must have,' 'should have,' 'couldhave' and 'would have,' known as the Moscow principle.
- Not enough consultation with real users and practitioners.
- Solving the 'problem' before you know what it is.
- Lacking a clear understanding and making assumptions instead of asking for clarification.

3.2.2Software requirement

- ➢ Spyder IDE
- ➢ Python (pandas,Numpy)
- ➢ HTML
- ➢ Flask
- ➢ OpenCV
- ➢ Imutils
- ➢ Geopy
- ➢ Requests
- ➢ Anaconda Distribution
- ➢ Windows OS

3.2.3 HARDWARE REQUIREMENTS



REQUIREMENT	**SPECIFICATION**
Operating system	Microsoft Windows UNIX Linux®
Processing	Minimum: 4 CPU cores for one user. For each deployment, a sizing exercise is highly recommended.
RAM	Minimum 8 GB.
Operating system specifications	File descriptor limit set to 8192 on UNIX and Linux
Disk space	A minimum of 7 GB of free space is required to install the software.

Table 1: Hardware Requirements

4.DESIGN

4.1 INTRODUCTION

The fast and reliable person detection (may be in any posture) from images, videos or from real-time webcam has been a goal of computer vision for decades. It has many applications including monitoring, gaming, human-computer interaction, security, telepresence, and even health-care. Person detection in images, videos and from the real-time webcam where a person may be in various postures like standing, sitting, lying, stooping etc. is the primary aim of this project.

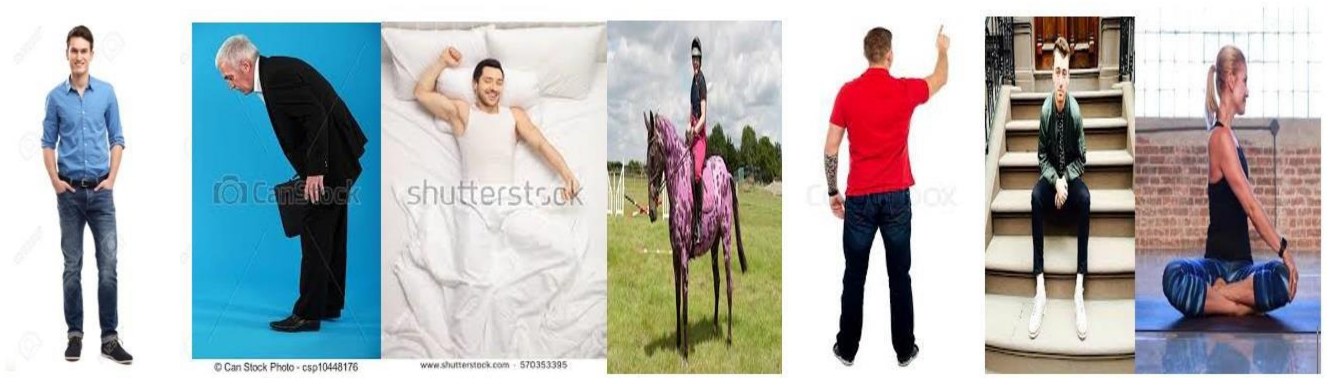


Figure 2: Person Detection in Various Postures

For this project, a collection of positive and negative samples is used. Positive samples are the objects of interest. And negative samples are those images having no object of interest. HOG (Histogram of Gradient) as the feature vector is used to train model. Samples of size 64*128 for both positive and negative are taken in which positive dataset contains persons in various postures (to train model). Various image processing libraries and machine learning algorithm such as sci-kit learn and OpenCV (which are the most powerful computer vision libraries) are used for implementation of the model and Linear SVM (Support Vector machine) is used to train model.

The HOG person detector uses a sliding detection window which is moved around the image. At each position of the detector window, a HOG descriptor is computed for the detection window. This descriptor is then shown to the trained SVM, which classifies it as either “person” or “not a person”.

The first major step is to prepare the positive and negative image dataset, after preparing image dataset, wherein positive samples-person should be in various postures as need and in negative samples anything except the object of interest i.e., images without the person. Mainly, there are three major steps to implement once the dataset is prepared: extraction, training, and detection. Training is the procedure to get the model which can easily classify the object of interest and detection is the algorithm which loads the model get from training and by applying some image processing techniques we could able to detect the object of interest form input image or video.

4.1 PROCEDURE

First and foremost, pre-processing step is to prepare the dataset for which Positive images and negative images are collected and then cropped of size 64*128. Generally, it is difficult to prepare the dataset manually. To prepare dataset, you first need to collect the images (Usually number of negative images are more as compare to positive images) and crop them in 1:2 ratio(generally)e.g., 64*128. Here the flowchart presents the whole process step by step.

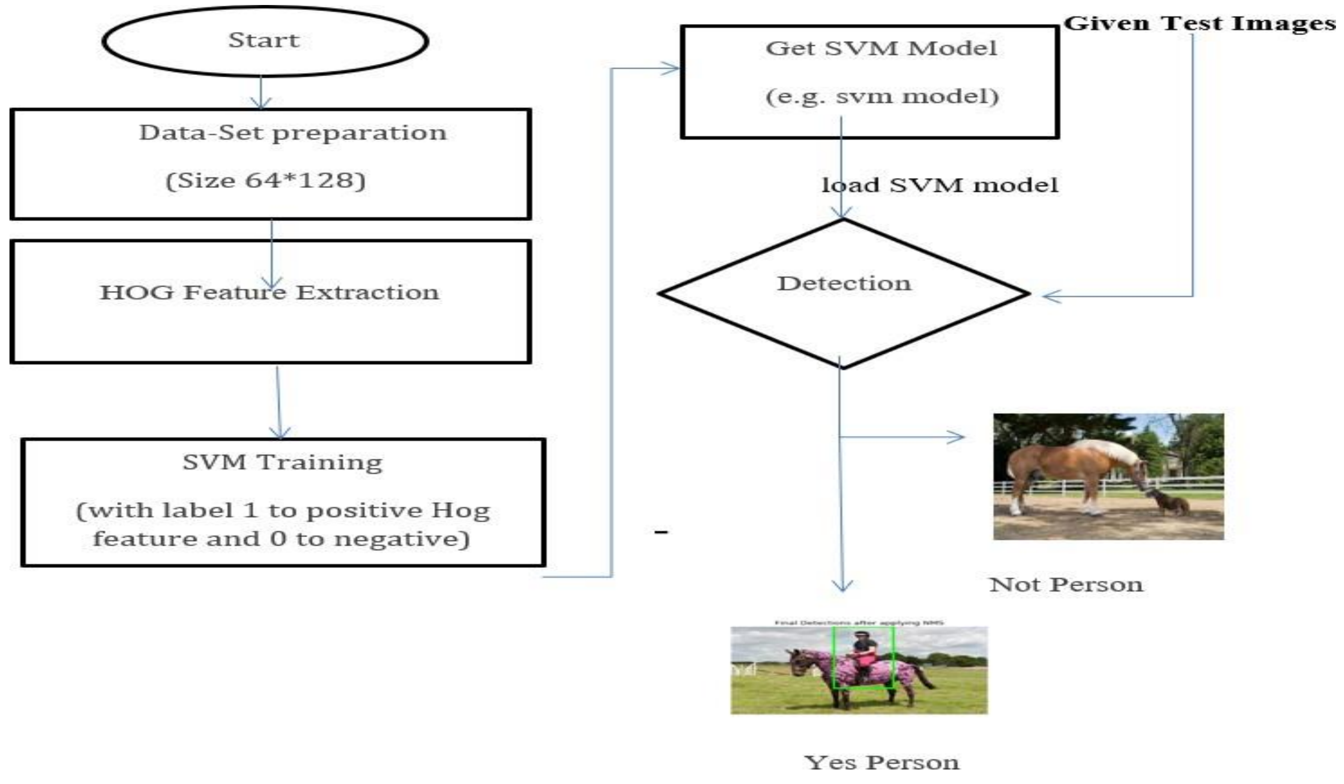


Figure 3: Flow diagram of project methodology

As our flow diagram proceeds next step is to extract the Feature from the prepared datasets. Feature extraction is the process by which certain features of interest within an image are detected and represented for further processing. It is a critical step in most computer vision and image processing solutions because it marks the transition from pictorial to non-pictorial (alphanumerical, usually quantitative) data representation.

The resulting representation can be subsequently used as an input to a number of pattern recognition and classification techniques, which will then label, classify, or recognize the semantic contents of the image or its objects. Hog feature is extracted from both positive and negative images.

After we come up with the feature vector, we are ready for our training. In training, label 1 is added to each positive feature and label 0 to each negative feature (which is extracted in the previous step). There is various machine learning algorithm e.g., the k-nearest neighbour, Support Vector Machine, etc. which can be used for training. Here the Linear SVM is used for training and we come up with a model which can be used in the detection algorithm to detect the person.

In detection, trained classifiers are used with a sliding window to localize persons in an image or video. Detection result can give more than one detection for a single person, that’s why the NMS algorithm (non-maximum separation algorithm) is used further which will give one detection for each person. And reducing the false positive by adjusting the various parameter is the interesting task here.

4.2 Algorithms and Techniques:

There are various algorithms used in this project in each stage as flow diagram proceeds.

4.2.1 Histogram Oriented Gradient (HOG):

It is one of the feature descriptors used in computer vision and image processing. Histogram of oriented gradients descriptor can be described by the distribution of intensity gradients or edge directions in an image. The image is divided into a small connected region called cells. A cell can contain several pixels and for each of the pixel, a histogram of the gradient is made. The descriptor contains histogram of the gradient of each and every pixel. For better accuracy the HOG is contrast-normalized, this is done by calculating intensity over a larger area several cell known as block, then this value is used to normalize all cells within that block. The normalized result gives better performance on variation in illumination and intensity. HOG descriptors have some advantages over other descriptors such as it is invariant to geometric and photometric transformations except object orientation. It is particularly suited for human detection in images.

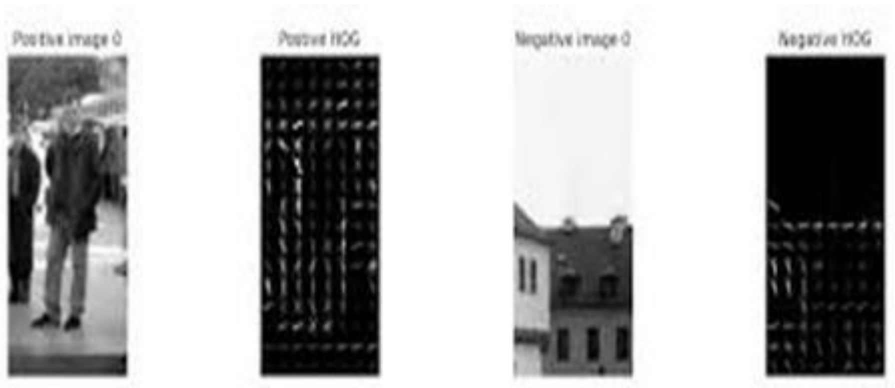


Figure 4: A visual demonstration of HOG Feature

4.2.2 Support Vector Machine (SVM) Classifier

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for either classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, classification by finding the hyperplane is performed that differentiate the two classes very well (look at the below snapshot).

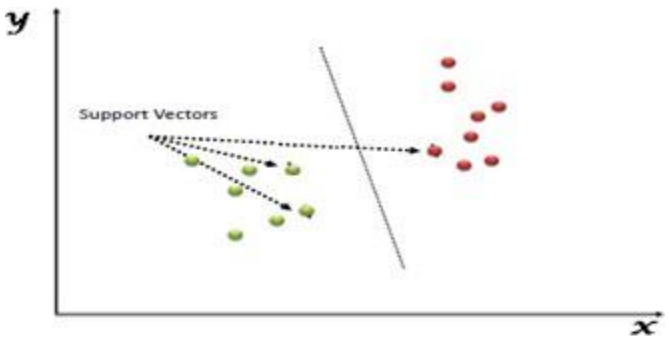


Figure 5: support vectors

Support Vectors are simply the coordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

In the linear classifier model, we assumed that training examples plotted in space. These data points are expected to be separated by an apparent gap. It predicts a straight hyperplane dividing 2 classes. The primary focus while drawing the hyperplane is on maximizing the distance from hyperplane to the nearest data point of either class. The drawn hyperplane called as a maximum-margin hyperplane.

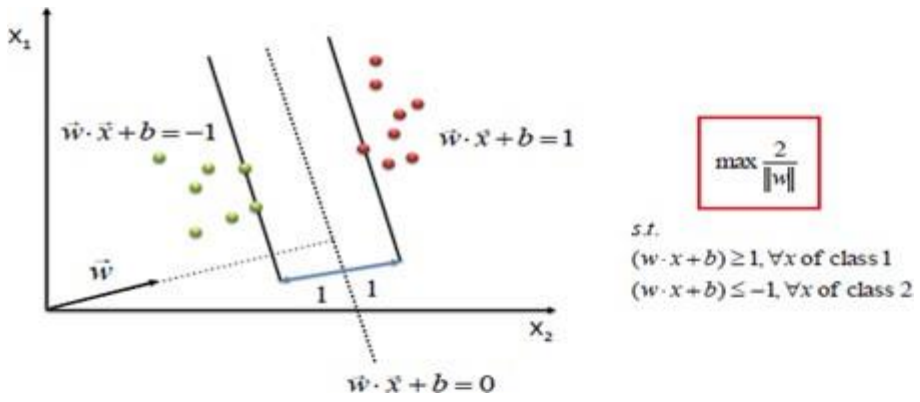

X_1
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\max \frac{2}{\|w\|}$
s.t.
$(w\cdot x+b)\geq 1, \forall x$ of class 1
$(w\cdot x+b)\leq -1, \forall x$ of class 2
$\vec{w}$
1
1
X_2
$\vec{w}\cdot\vec{x}+b=0$

Figure 6: A demonstration of Support Vector Machine

4.2.3 Sliding Window & Non-Maximum Separation Algorithm for Object Detection

In the context of computer vision (and as the name suggests), a sliding window is a rectangular region of fixed width and height that "slides" across an image, such as in the following figure:

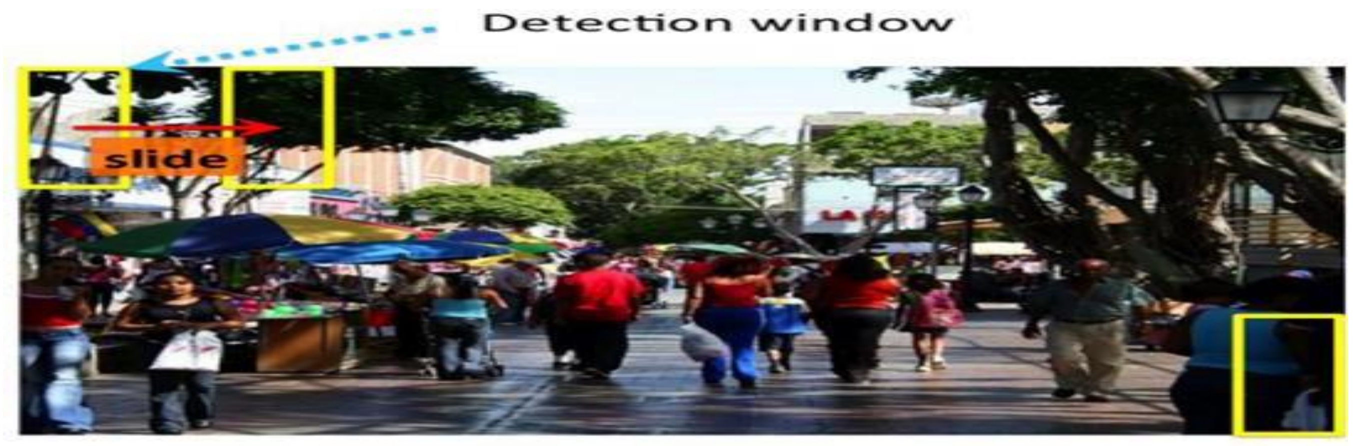


Figure 7: Example of the sliding a window approach

For each of these windows, normally the window region is taken and an image classifier is applied to determine if the window has an object that interests us — in this case, a person. Combined with image pyramids image classifiers can be created that can recognize objects at varying scales and locations in the image. These techniques, while simple, play an absolutely critical role in object detection and image classification.

When using the Histogram of Oriented Gradients descriptor and a Linear Support Vector Machine for object classification you almost always detect multiple bounding boxes surrounding the object you want to detect. Instead of returning all of the found bounding boxes first apply non-maximum suppression to ignore bounding boxes that significantly overlap each other.

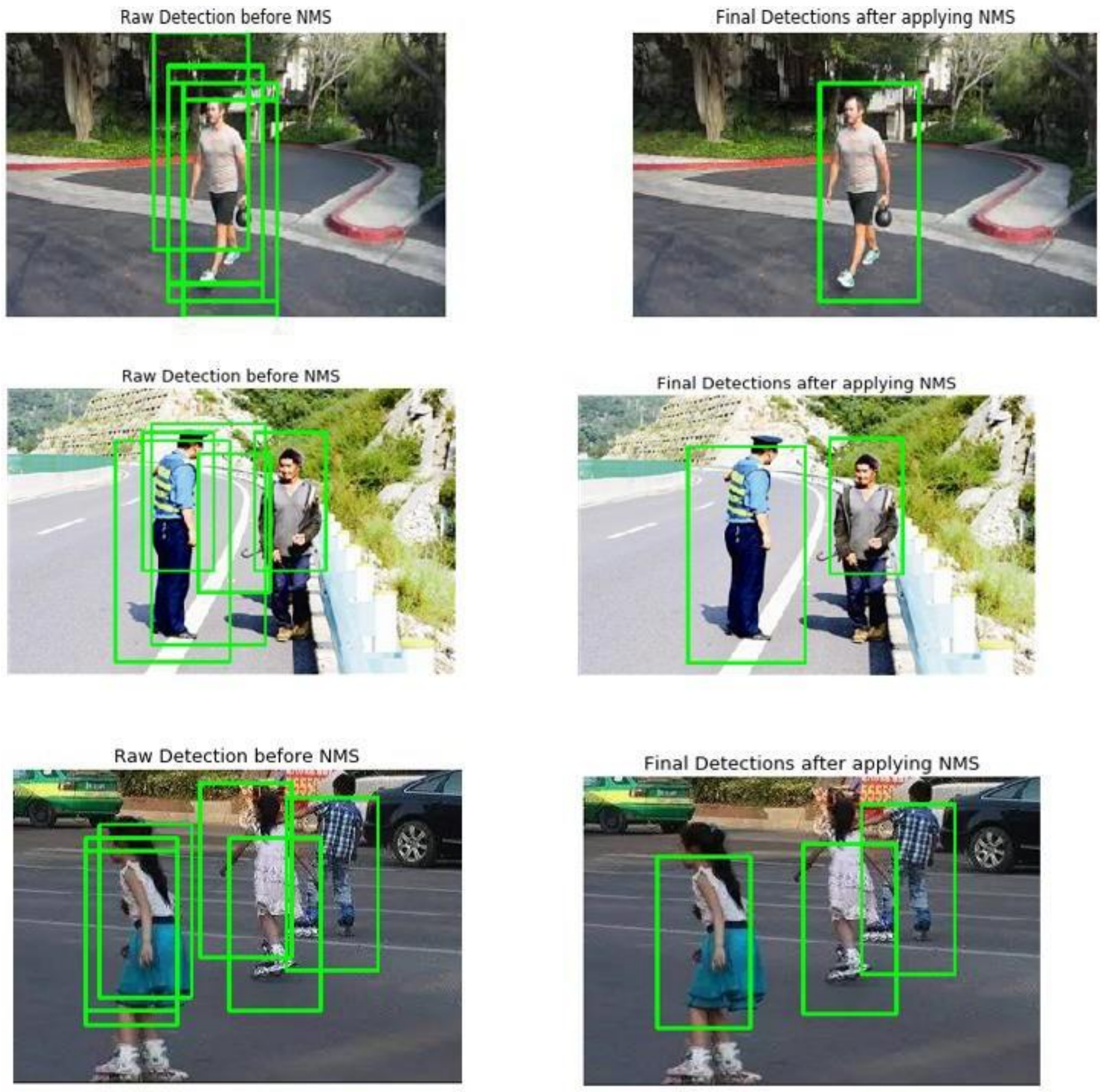

Raw Detection before NMS
Final Detections after applying NMS
Raw Detection before NMS
Final Detections after applying NMS
Raw Detection before NMS
Final Detections after applying NMS

Figure 8: Some of the detection result before NMS and after NMS algorithm