AJE Fire Detection System

Version 1.0

10/17/2018

AJE

Bellevue College Department of Computer Science Anis Beyzaee, Joey Colombi, Ephraim Scarf Alfred Nehme

Revisions Page

Date	Version	Changes
10/17/2018	1.0	Creation of document
11/30/2018	1.1	Block Diagram Update Functional requirements added
3/10/2019	1.2	Component/ Diagrams updated

Table of Contents

- 1. Introduction
 - 1. Purpose
 - 2. Scope
 - 3. Definitions, Acronyms & Abbreviations
 - 4. References
 - 5. Overview
- 2. Overall Description
 - 1. Product Perspective
 - 2. User Characteristics
 - 3. Constraints
 - 4. Assumptions and Dependencies
- 3. Specific Requirements

- 1. External Interfaces
- 2. Functional Requirements
- 3. Performance Requirements
- 4.Software System Attributes

1 Introduction

1.1 Purpose:

Forest fires are a major concern in the present day global ecosystem, causing harm in ways such as destroying timber, killing wildlife, and polluting the air. Our goal in developing a Forest Fire Detection System for EarthNow, a satellite imaging system, is to not prevent forest fires from occurring, but to detect when they first start and notifying authorities in order to minimize the damage.

1.2 Scope:

Given that so little is known about what the EarthNow API will look like, we decided to focus on building just a fire detection module that would eventually become a part of our complete system. The finished detection module would take satellite images as inputs, and output 'true' or 'false' for whether or not the provided image contains a fire. The detection module would use an artificial neural network to analyze images, and so the ANN would need to be developed and refined by taking training data as input. Training data would consist of images that have fires and images that don't, plus the true classification for each image, that is, whether or not a given image actually has a fire. In order to streamline the process of testing and training the ANN, our application will have a GUI, which will allow us to easily input training and testing data into the application.

1.3 Definitions, Acronyms & Abbreviations

- API: Application Programming Interface
- ANN: Artificial Neural Network
- AJE: The name of our system
- GUI: Graphical User Interface
- Mentor: Instructor guiding us with the project

- Greyscale
- Backpropagation
- ANN Algorithm
- 1.4 References
 - Coldewey, Devin (18 April 2018). "EarthNow promises real-time views of the whole planet from a new satellite constellation". TechCrunch.

1.5 Overview

Section 1 of this document is the introduction, which contains a broad description of the project and is the recommended starting point for reading. Section 2 gives more specific information about the context surrounding our project and how it will be used. Section 3 describes the specific requirements that we feel our project must be able to fulfil.

2. Overall Description

2.1 Product Perspective

The AJE fire detection system is ultimately not self-contained. While the system will be able to detect fires in images that are manually given to it, our goal is to have the system analyze images that are received automatically from live satellite feeds. In order to do this, our system would have to interact with the EarthNow API, which is currently non-existent. Below is a block diagram outlining the interconnections of our system and external systems.

2.2 User Characteristics

- United States Forest Service: Limited high-level technical expertise, but can access a computer and can use and comprehend the application and the API it is on.
- 2.3 Constraints
 - The availability of the EarthNow API
 - The frequency of updated satellite video frames
 - Access to existing images for training
- 2.4 Assumptions and Dependencies
 - Be able to establish a safe and secure connection
 - There is access to reliable pictures of forests
 - There is access to a reliable and ample amount of training pictures to build our training set
 - There will be access at some point to the EarthNow API
 - The internet will be accessible to find our data
 - There will be no hardware limitations

- 3. Specific Requirements
 - 3.2 Functional Requirements
 - The system shall take training sets of fire images and recognize whether there is a fire or the beginning of a fire (smoke) or if there is no fire
 - The system shall send a notification to the admin when it recognizes a fire in the image given
 - The system shall take real inputs of satellite images and determine whether the image contains a fire or not
 - The system shall be able to take images with a variety of sizes and convert it to one fixed image to be used throughout the application

- The system shall run as a service on either a Windows or Linux operating system.
- In the event that the computer on which the system is running shuts down, the system service should start automatically when the computer restarts
- 3.3 Performance Requirements
 - The system shall be able to analyze the image given has a fire or not in less than five minutes
 - The system shall have an accuracy rate of at least 90% when attempting to detect if a given image has a fire or not
- 3.4 Software System Attributes
 - Portability: The system shall be compatible with many API's
 - Testability: Putting in more training data into the system will improve the accuracy of its ability to detect a fire