Functional Specification

Andrew Cullen – 17378471
Eoin Joseph McKeever - 17436202

Table of contents

1) Introduction 2
    a) Purpose 2
    b) Overview 2
    c) Business Context 2
    d) Glossary 3
2) General Description 4
    a) System Functions 4
        i) Upload Video 4
        ii) Deriving Tabs from Video 4
        iii) Viewing Tabs 4
        iv) Downloading Tabs 4
    b) User Characteristics and Objectives 4
    c) Operational Scenarios 5
        i) Video Upload 5
        ii) Video Conversion 5
        iii) Tab Viewing 5
        iv) Tab Download 6
    d) Constraints 6
        i) Training Dataset 6
        ii) Time 6
        iii) Financial 6
        iv) Neural Network Accuracy 6
        v) Users Requirement
3) Functional Requirements 7
    a) Mask Regional based Convolutional Neural Network 7
    b) OpenPose 7
    c) Doremir 8
    d) Image Processing 8
    e) Tab Formulation 9
4) System Architecture 10
    a) Mask Regional Neural Network Model 10
    b) General System Architecture Diagram 11
5) High-Level Design 12
    a) High-Level Context Diagram 12
6) Preliminary Schedule 13
7) Appendices 14

Introduction

a) Purpose

This document’s purpose is to describe the problem and the proposed application that tries to solve it, Tabulator. We will outline the implementation of the application at an abstract level and its intention, design, and scope.

b) Overview

Guitar tabs are a key part in a guitar player’s learning experience. They are also a useful tool in transcribing one's own play for future reference. Digital versions of guitar tabs are now more popular than hand sketched as they can be viewed on a phone, laptop or printed out and can be shared with ease.

The project is to create a web app that will take a video of guitar play as input and create guitar tabs from the audio and visuals of said video. These guitar tabs will be displayed to the user and be available to download for the user to keep. Allowing them to learn or remember the song that was played. The system will solve the problems of creating tabs from the users own guitar play for the purpose of easily reproducing the play later or for creating easy to learn tabs from videos of guitar covers and tutorials available on the internet. This will also be a very useful tool for guitar teachers (both online and personal) as one can simply record themselves playing instead of manually writing up a tab.

We will be using machine learning to implement this project. Three different neural networks, two readymade, OpenPose and Doremir to recognise the player’s hand and the notes being played. Then we will create our own R-CNN, a regional convolutional neural-network that will identify the region of the image that is the guitar’s fretboard. With this information we will be able to create a guitar tab with exact fingering.

c) Business Context

The system would not have many business applications other than possibly advertising revenue online. The system could also be made part of a paid application for guitar players.

d) Glossary

Machine Learning - An application of artificial intelligence that allows systems to learn without being explicitly programmed. This is done through developing programs that can access data and learn for themselves.

Guitar Tab - A form of music notation for guitar that shows each string on the fretboard and where each finger is placed on the fretboard when a note or chord is played.

Sheet Music - A form of music notation that shows each note that is played and the timing of when it is played.

Fret - A rung on a guitar's fretboard which the player's finger can be placed to create a note when the string is struck.

Fretboard/Fingerboard - The part of the neck of the guitar where the player's hand is placed to press down the strings to play certain notes.

Fingering - The arrangement of the player's left hand to press down the strings to achieve the correct note or chord.

Neural Network - A computer system vaguely modelled on the human brain and nervous system using systems of neurons which are connected using weights that adjust as learning proceeds.

Convolutional Neural Network (CNN) - A convolutional neural network (CNN) is a specific type of artificial neural network. Generally used for image processing problems.

Region based-CNN(R-CNN) - A Region based - Convolutional Neural Network (R-CNN) is a specific type of artificial neural network that makes use of CNN alongside the selective search algorithm which extracts regions of interest(ROI) from the input images.

Application Program Interface (API) - A set of protocols to mandate how software components (usually in a client-server relationship) interact. In essence, a way to access some resource in a restricted/structured manner.

OpenPose - A python library which can be used to recognise parts of the human body in an image and their movements.

Doremir - An audio processing API based on a neural network that can be used to create sheet music from audio files of the playing of musical instruments.

Segmentation - The process of assigning label to every pixel

Image Recognition - Is the process a piece of software takes to identify objects, places, people, writing and actions in images.

Fast R-CNN - A faster version of a R-CNN, it is coded slightly differently.

Mask R-CNN - Adds instant segmentation to Fast R-CNN.

General Description

a) System Functions

Upload Video
The user can upload a video of their guitar playing in the .mp4 format. Once this is uploaded this will be separated into audio; an mp3 file and the video will then split into relevant frames where notes are being played for analysis by our R-CNN and OpenPose.

Deriving Tabs from Video
The audio from the video will be taken and converted into raw data for the Doremir API to take it and convert it to a music XML file. This will contain all the information of a piece of sheet music; the notes being played as well as when they are played. With this information we will then be able analyse the video, or more specifically, the frames of the video when notes are being played. These frames will be processed using OpenPose and our R-CNN, the R-CNN will find the region of the image where the fingerboard lies as well as it is orientation (where is the top/bottom). We will then use OpenPose to pinpoint the position of the hand on the fingerboard. With the position of the hand relative to the fretboard we can derive which fret of the guitar is being played within an acceptable margin of error. We need this information as while we can produce sheet music from just the audio, to make tabs we also need the hand position because any note can be played in up to five different places on the guitar. All of these are evenly split at even intervals up the length of the fretboard. As we now have all this information, we can then construct the guitar tabs.

Sidenote - There are many intricacies that can arise when making guitar tabs due to the complexities of the instrument. Due to time constraints and other factors that may arise, we may not be able to generate perfect guitar tabs. There are techniques such as string bending, vibrato, sliding etc. that are part of many guitar pieces which are denoted in tabs. The project will be an exploration into how far we can get in terms of accuracy and these and other stretch goals.

Viewing Tabs
The user will be able to view the finished guitar tabs on the webpage when the processing is finished. The .tab file is a simple text file and as such can be displayed with ease on the web page in a container.

Downloading Tabs
Under the container with the finished tabs will be a button to download into a text file. The user will be shown an input box for their file name to be downloaded onto their local computer.

b) User Characteristics and Objectives

The users of this system will be made up of guitar players. As such these users will be expected to know what guitar tabs are and how they are used. No other prior knowledge is expected of the user, as such we will focus on a simple and intuitive user interface. We could split the user base into guitar teachers and guitar players. Teachers will want to be able to

show and share the tabs with their students easily. Players will want to convert their play and the play of others into tabs for their own learning or recollection.

c) Operational Scenarios

Scenario 1: Video Upload

System state: The user loads the web page and is shown the video upload button.

Action: The user clicks the upload button and is shown a dialogue where the user selects an .mp4 file from their local upload. The video will then be posted to the server.

Result: The user will be prompted with a “convert” button if the file is accepted and shown an error message if an error has occurred such as the file was in the wrong format.

Scenario 2: Video Conversion

System state: The user has uploaded the .mp4 file to the server and the user clicks the “convert video to guitar tabs” button.

Action: The video is then processed, and the user is presented with a loading icon while the video is processed server side. First the audio will be taken from the video and sent to the Doremir API, here we get the notes played and the times for which we take frames from the video. The video frames will then be sent to the neural networks and tabs will be derived from these frames processed between OpenPose and the R-CNN and the audio processed by Doremir. These processes are interpreted using logical statements to write the resulting tabs.

Result: The resulting tabs are displayed to the user with an option to download.

Scenario 3: Tab Viewing

System state: The user has successfully converted a video to guitar tabs and the user is presented with the tabs in a container they can review before they click the download button below.

Action: The user can examine the tabs in the container. This container is essentially the tab file in a scrollable box. This will make up most of the webpage as it is desirable for the user to be able to play through the whole song without touching the mouse or keyboard.

Result: The user has the full tabs from the video they have uploaded which they can view either to play with them displayed on the site or simply for review before downloading them.

Scenario 4: Tab Download

System state: The system is in the same state as scenario 3: Tab Viewing, but the user clicks the “download” button below the tabs.

Action: Once the user clicks the “download” button the user is prompted with a text field where they name the file. The tabs will then be downloaded in a text file from the server to the user’s computer.

Result: Once the file has downloaded successfully the user will be shown a success message but will remain on the same tab viewing page.

d) Constraints

Training Dataset constraints
For the neural network problem there is no dataset that can be pulled from the web to use directly that is known. Due to this one must be created. This can be very time consuming especially when all the different ROI’s must be identified in each training image manually. Supervisely will be used which is a service which helps to prepare neural network datasets.

Time Constraints
The application is to be submitted by the 6th of March 2020. Given the different degrees of accuracy that can be reached with the project and the different features that could possibly be introduced, the goals may need to be adjusted as the project proceeds. This will be dealt with on a rolling basis, as the schedule proceeds.

Financial constraints
As this project is not fully scaled to a business model, the use of certain tools will be limited, as these tools offer a less effective and polished community version and a paid version as this is not a profitable business model the resources are not available to purchase these paid versions of the tools. Furthermore, Doremir is a paid API and there is a cap to the amount of times we can call it.

Neural Network Accuracy constraints
When it comes to neural networks even if you train on an excellent dataset and spend a large amount of time tweaking and improving the network it still may not reach an extremely high level of accuracy, say 98% or so. The level of accuracy that is achieved could act as a constraint on our project.

Users requirement constraints
Each user’s personalised music tab results must be accurate, or the app will not have done its job. Making sure that each tab sheet is outputting the correct result is an important constraint on our application.

Functional Requirements

a) Mask Regional based Convolutional Neural Network

Description: The Neural Network that will carry out the image recognition task is called a Mask Region based Convolutional Neural Network (Mask R-CNN). The structure of this can be seen in figure 4.1. A mask R-CNN works similarly to a Fast R-CNN. In the fact that Fast R-CNN runs the convolutional neural network once on the whole image. While an ordinary R- CNN will individually compute the neural network features on what could be up to 2 thousand ROI. Avoiding this vastly increases the speed. A key difference that differentiates a Mask R-CNN to a Fast R-CNN is that a Mask R-CNN replaces the Python method ROIPooling with a new Python method called ROIAlign, which can represent fractions of a pixel, this helps improve accuracy when attempting segmentation. Mask R-CNN uses instance segmentation. This is done by adding a binary mask to every pixel. The mask represents whether the pixel refers to the object we are interested in. With this mask segmentation can be done very quickly. This is done after the ROI Align method is called on the feature map. The plan is to use this Mask R-CNN to classify the position of the top and the bottom of the guitar neck and pull the pixel coordinates for both the top and bottom of the neck. Using a distance formula, the length of the neck of the guitar can be computed using the pixel coordinates outputted. The pixel coordinates may need to be normalised.

Criticality: This is a critical part of the success of the application as the output the user receives relies heavily on the result produced by this neural network.

Technical issues: Achieving a high degree of accuracy can be quite difficult. Especially if a dataset is not constructed to perfection which is a possibility when constructing a new dataset.

Dependencies with other requirements: The Mask Regional based Neural Network has a dependency with the Tab Formulation functional requirement which is discussed below. It supplies in pixel terms the length of the neck of the guitar which will be used to formulate the tabs. It also depends on the Doremir to feed it the correct frame to investigate. The Neural Network also depends on the image processing functional requirement mentioned below to create and process its training dataset and to process the unknown images it is fed by the user.

b) OpenPose

Description: The project will make use of the Python's OpenPose API package. What OpenPose has the ability to do is recognise human body parts i.e. a hand. The plan is to run this over the input video. What it has the ability do is give back the pixel coordinates of where the hand is situated on the guitar neck and this will be used to help derive the tabs. With the hand coordinates paired with the knowledge of the length of the neck in terms of pixel coordinates the fret the hand is situated on can be computed through some simple mathematics.

Criticality: The derivation of the final output depends heavily on this process being successful. Therefore, it is quite a critical part of the application.

Technical issues: One issue is that OpenPose struggles to identify overlapping body parts so if another hand was in the shot for some reason issues may arise. But in theory this should not happen. It is compatible with the TensorFlow API which will be used to build the Neural Network model which is good.

Dependencies with other requirements: OpenPose is dependent on Doremir to feed it the correct frame to investigate. It also has a dependency with our Tab Formulation functional requirement which is discussed below. It feeds it the pixel coordinates of the location of the hand on the guitar neck which will be used to evaluate the output.

c) Doremir

Description: Doremir is a musical audio recognition API which can receive audio files and convert them to sheet music (For our purposes). Doremir will be used to derive the frequencies which occur in the audio and when they occur. For the purpose of this project a mp3 file will be posted to Doremir and a music XML file will be returned.

Criticality: This is critical to the entire functionality of the system. We know which notes are played through Doremir. The image processing also relies on this to provide the time to choose the frames to analyse.

Technical issues: There may be intermittent technical issues when trying to access the API from initial testing. Sometimes the responses may be slow or timeout, although this is very rare. The format in which we receive the data from Doremir will also have to be processed and reformatted for our purposes.

Dependencies with other requirements: Tab formulation and image processing, OpenPose and the neural network all depend on Doremir to provide a time stamp for the images they will analyse.

d) Image processing

Description: The images must be pre-processed before being fed into the Neural Network as training images or for analysis. Supervisely will be used to manipulate the training dataset. From images of guitars region-based classifications of the desired ROI will be applied to specific parts of the guitar i.e. neck of guitar and head stock. Images will be resized and reshaped within the Python code.

Criticality: This is an important part of our project because without a correctly formatted dataset the Neural Network will struggle to learn efficiently. Also, if the images aren’t resized and reshaped issues will arise.

Technical issues: The technical issue here is that in Supervisely for each image you have to manually crop the ROI in each image since unlimited time is not available to complete this project this could be an issue that could cause the dataset to be smaller than is suitable.

Dependencies with other requirements: Image Processing has a dependency with the Neural Network because the Neural Network needs to be fed the images after they are processed.

e) Tab Formulation

Description: Tabs are required to be formulated once all the required data is gathered. This will involve grabbing the result produced by the Doremir API, OpenPose API and the Neural Network results. Doremir will supply what notes are being played, while the combination of the OpenPose API results and the Neural Network results which will both be pixel coordinate points. As described earlier the length of the neck of the guitar can be derived using the distance formula and the coordinates of the top and bottom of the neck. Now pair this with the coordinates of the hand to find what fret the hand is situated on using some simple mathematics. So now whatever note is being played is known and the position on the guitar the note is being played is known. Now the tabs can be derived using some Python code.

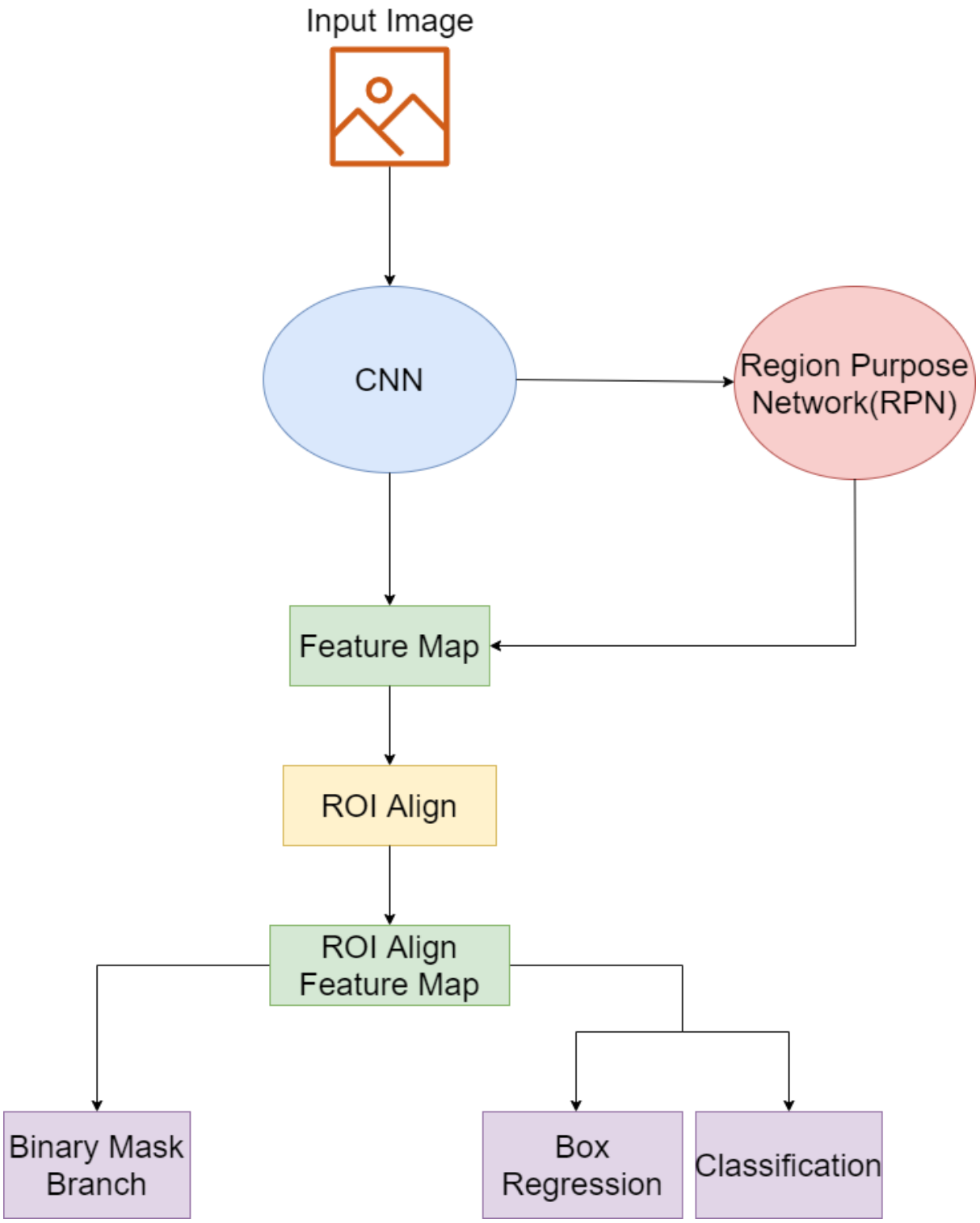
Criticality: This section is essential for the success of the project if an accurate result is not produced here this will arise a serious issue

Technical issues: The only technical issues that could occur here is if the input given is not accurate. It is essential the systems that produce the input are tested extensively.

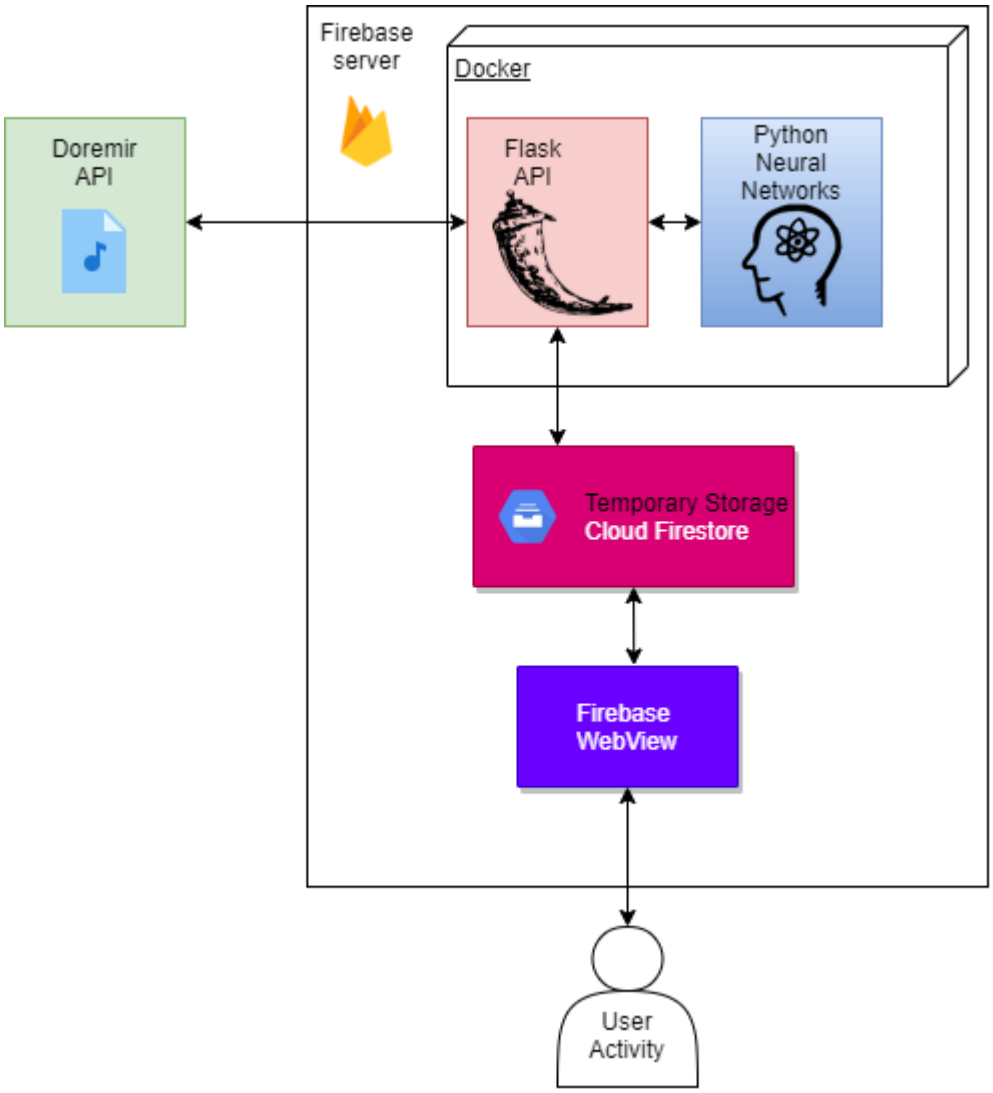
Dependencies with other requirements: The formulation of the tabs depends heavily on many other functional requirements i.e. OpenPose, Doremir and the Neural Network. It uses the output of all these to compute the tabs.

System Architecture

a) Mask Regional Neural Network Model

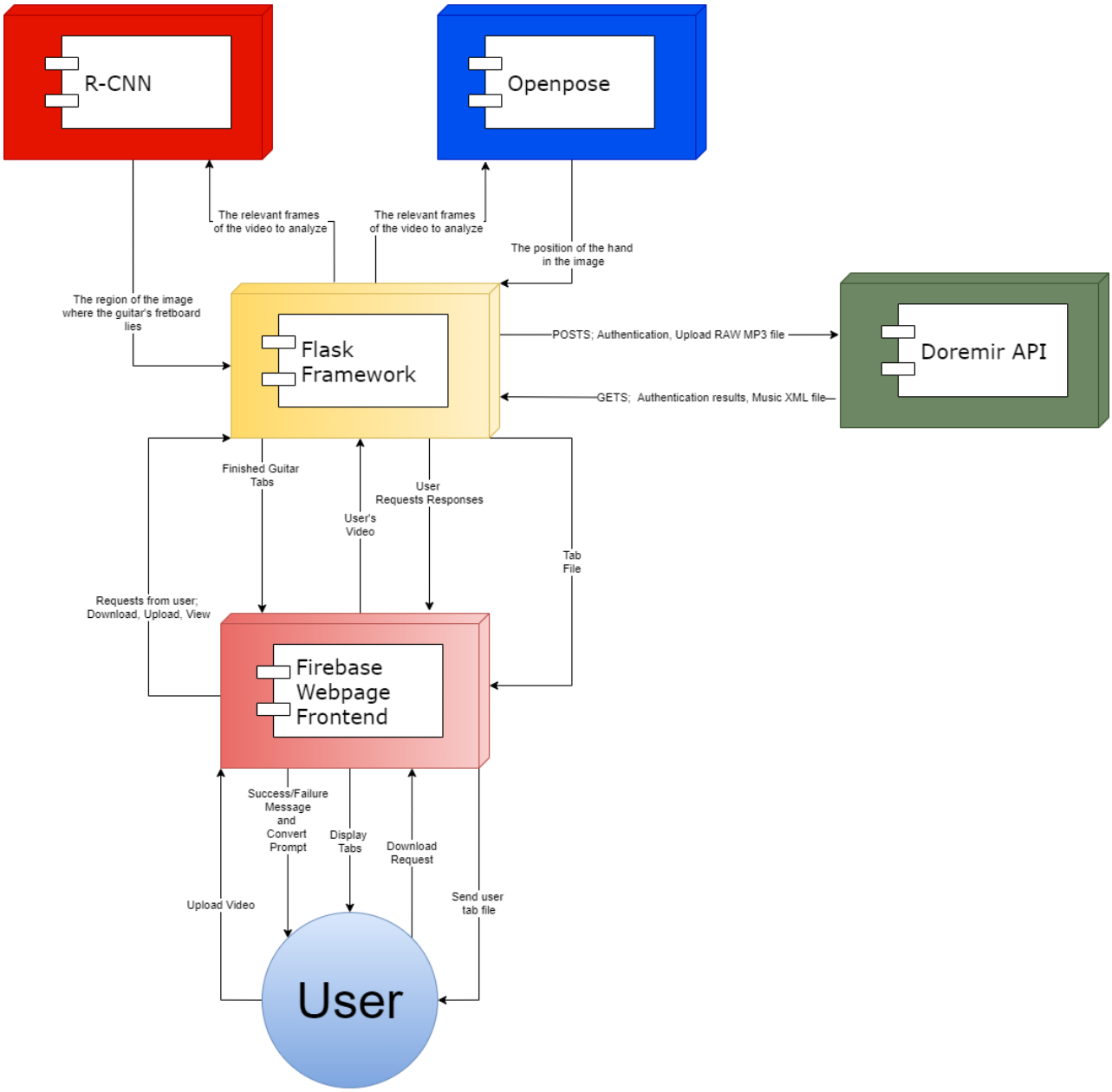


b) Architecture Diagram

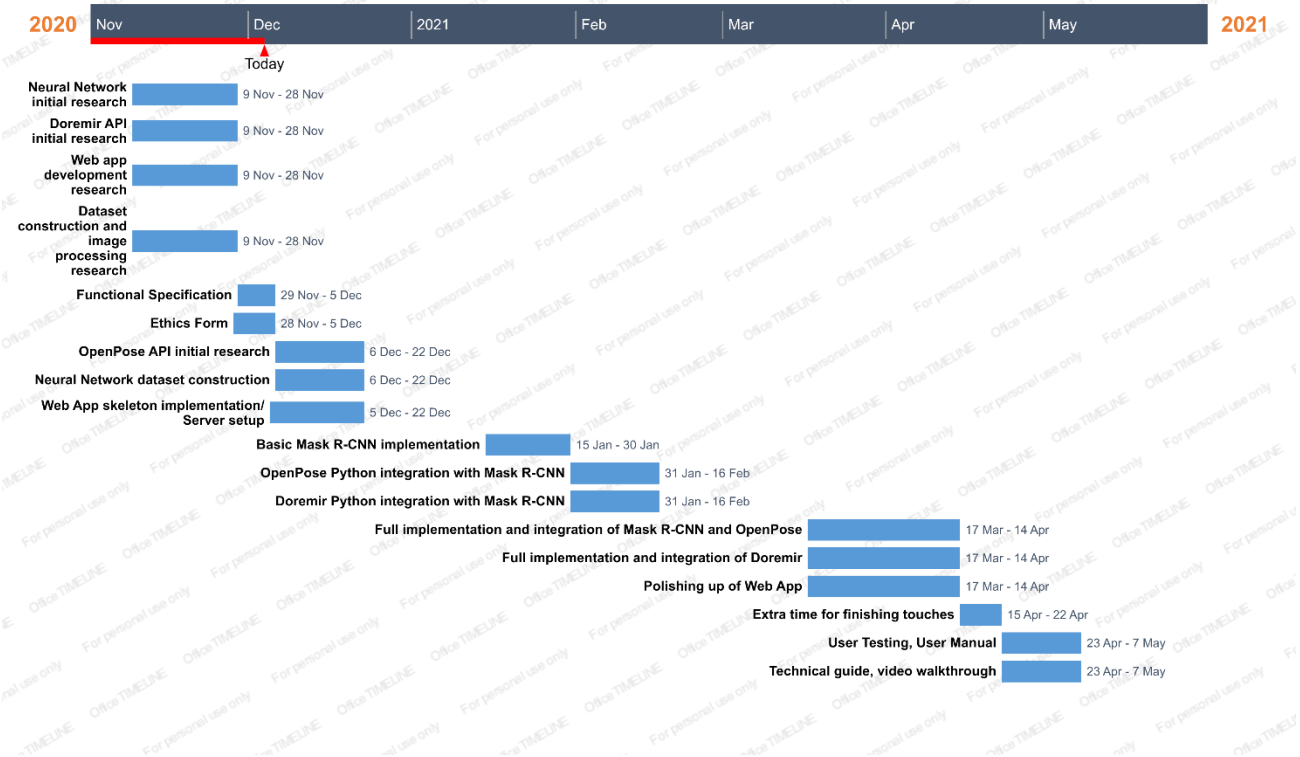


High-Level Design

a) High-Level Context Diagram



Preliminary Schedule



Appendices

Supervisly: https://supervise.ly/

Flask: https://flask.palletsprojects.com/en/1.1.x/

Firebase: https://firebase.google.com/

Doremir: https://doremir.com/

OpenPose: https://github.com/CMU-Perceptual-Computing-Lab/openpose