SANTA CLARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 3, 2022

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Austin Johnson

Carlos Mercado

Sabiq Khan

ENTITLED

Improving Diversity in Journalistic Sources with Computer Vision

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

BACHELOR OF SCIENCE

IN

COMPUTER SCIENCE AND ENGINEERING

Yi Fang (Jun 13, 2022 13:53 PDT) Jun 13, 2022

Thesis Advisor(s) (use separate line for each advisor) date

N. Ling (Jun 13, 2022 14:29 PDT) Jun 13, 2022

Department Chair(s) (use separate line for each chair) date

Improving Diversity in Journalistic Sources with Computer Vision

By

Austin Johnson,

Carlos Mercado,

Sabiq Khan

Submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Science in Computer Science and Engineering

School of Engineering

Santa Clara University

Santa Clara, California

June 9, 2022

Improving Diversity in Journalistic Sources with Computer Vision

Austin Johnson,

Carlos Mercado,

Sabiq Khan

Department of Computer Science and Engineering

Santa Clara University

ABSTRACT

News is a constant part of our lives, and has a significant impact on how we see the world. Simply put, journalistic sources are not diverse enough, and typically are only those in the majority. Newsrooms have made passionate declarations about wanting more diversity in their news sources. Our project aims to help with that. Building on top of the DEI toolkit, we aim to help newsrooms increase the diversity of their sources by analyzing their sources using a google search to pick an image and facial recognition software to determine the gender and race of the image.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Although there are more ways to get news than ever, the vast majority of sources within an article come from a select population and do a poor job of including underrepresented communities. Increasing the diversity of the sources we use and the people we feature is the first and most significant step in creating journalism that paints a more complete picture and is more relevant to audiences. Under-serving minority groups through a lack of diversity also has a knock-on effect on the trust in the media and unwillingness to speak to the press. By not having a wide range of backgrounds represented in news sources, journalists miss out on stories in worlds they may not belong to and certain topics they might not be familiar with.

If this continues to be the case, important issues relating only to underrepresented communities will not be heard or solved. One of the issues is that many large newsrooms who want to have a diverse set of sources do not have an efficient method to do it. Our project would provide newsrooms with a tool to solve that problem.

1.2 Solution

By providing a detailed, visual breakdown of sources, journalists will be provided with a better idea of where their sources are coming from and how diverse they are. We plan to solve this problem with computer vision and natural language processing to provide an automated way to suggest information such as the race or gender of the sources themselves from a given article. Leveraging Stanford's Core Natural Language processing server, there is already a system in place which identifies quotes and gives a suggested gender of the source.

We want to build on this system by providing a race suggestion, along with adding or reducing confidence in the gender and race suggestions utilizing computer vision. We want to scrape the internet for images of the source's name and make a suggestion of their race and gender to, as said before, build or reduce confidence in the initial suggestion. The intent of this project is not to profile people based on their race or gender, but to provide suggested information which can help newsrooms solve the problem of a lack of diversity in their sources.

Chapter 2

Background Information

2.1 DEI Toolkit

The DEI Toolkit [7] is a wordpress plugin newsrooms use to analyze and display the diversity of their sources. This toolkit already exists and is the project we plan to build on top of.

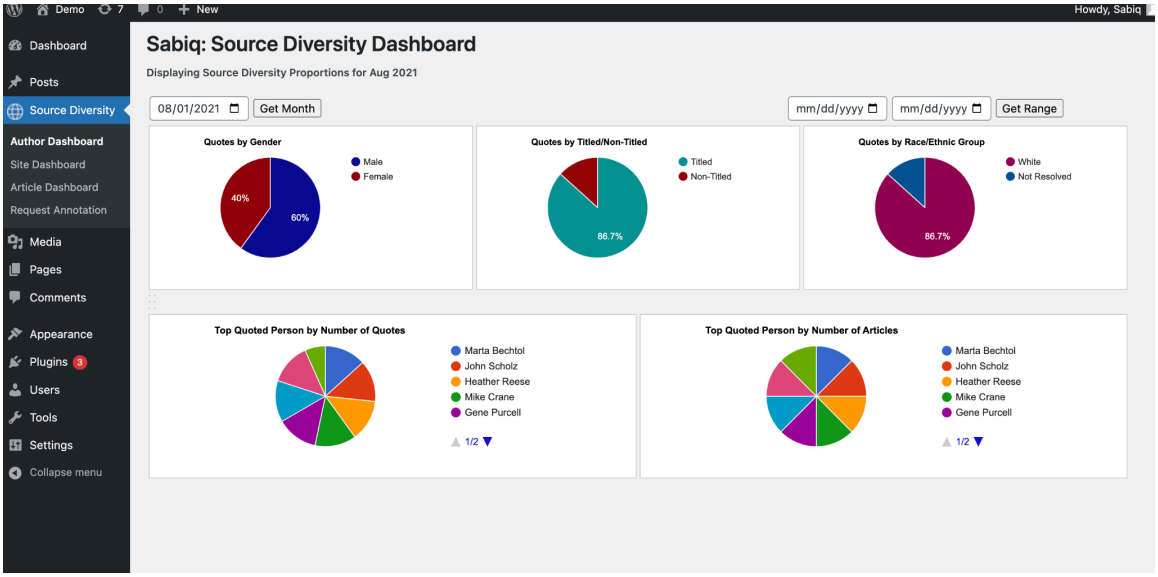

Figure 2.1 DEI Toolkit Site Dashboard page

Here we can see the diversity of the newsroom's sources based on a time frame. In this example, it's for the month of August.

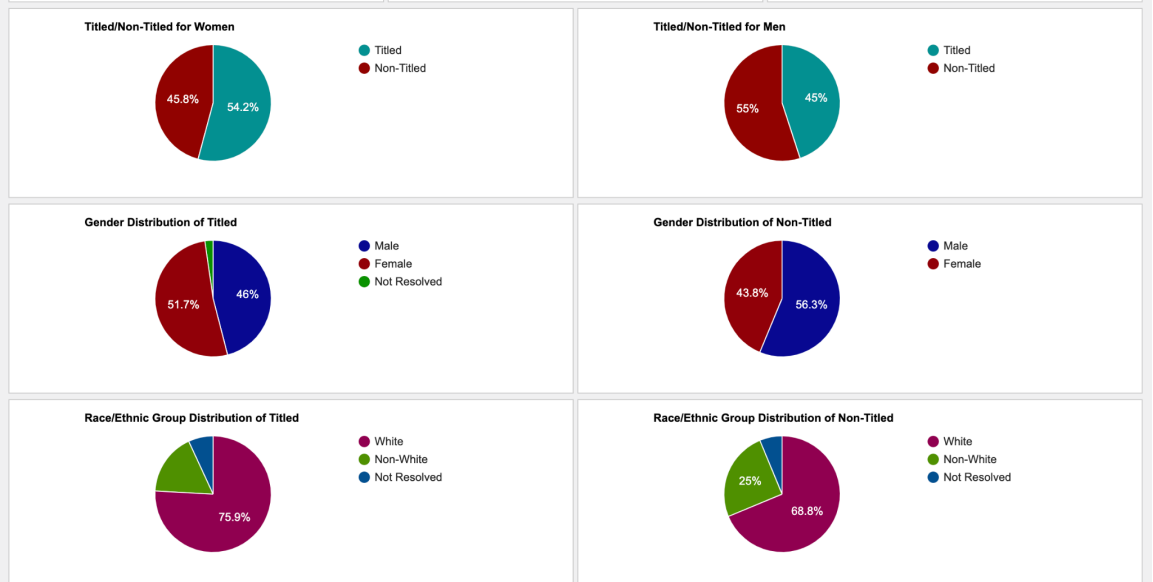

Figure 2.2 DEI Toolkit Second Level Statistics

Here we can see the different insights the DEI toolkit provides, including second level statistics such as the gender distribution of titled quotes, the title distribution for a certain gender, etc.

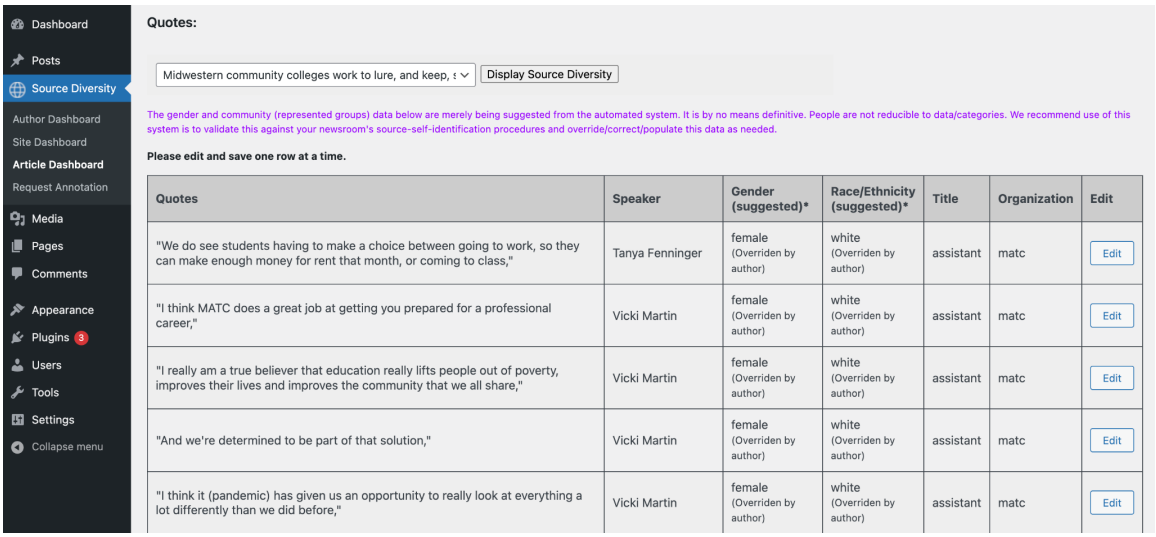

Figure 2.3 DEI Toolkit Article Dashboard

Here is the article dashboard page, where journalists are redirected after they press the ‘Get Source Diversity’ button on their article draft. The page displays a table with all of the quotes, along with information about the quote. The system also includes an override feature, because as effective as the machine is at predicting diversity metrics, it will never be one hundred percent accurate, and people cannot be minimized and simply categorized incorrectly as what a computer may think they are.

2.2 Current Issues with the DEI Toolkit

The DEI Toolkit is generally effective in predicting an individual’s gender and race based off of a name, but there are many cases where it struggles. An existing pitfall of the current system is its inability to distinguish between gender neutral names. For example, the current NLP system may categorize Alex Morgan, the famous women’s soccer player, as a man. The natural language processing cannot correctly distinguish between a man and a woman for a gender neutral name, but an image will clearly show that Alex Morgan is a woman.

Another issue with the current system is that it struggles with Black and White names frequently, along with names that are uncommon. Austin Johnson is a white Computer Science and Engineering student at Santa Clara University. Austin Johnson is also a Black professional football player for the Tennessee Titans. Based on the text alone, the system would not be able to distinguish between the two individuals. An image search with a title included will return a more accurate representation of who we’re looking for, and will be able to identify and categorize the individual.

Chapter 3

Requirements

Using the problem statement above, functional requirements, non-functional requirements, security requirements, and design constraints have been made.

3.1 Functional Requirements

- The system will take a text-based query from the user
- The system will get an image result from Google Images
- The system will then return a race and gender based on the image with a confidence interval for both.

3.2 Non-Functional Requirements

- The system should work with non-gender specific names.
- The system should be able to complete within a reasonable time.
- The system should integrate smoothly with the DEI Toolkit.
- The system should have an option for text based interface.
- The interface should be simple to the user.

3.3 Design Constraints

- The system must be able to connect with the DEI Toolkit.
- The system must be completed by May 2022.

Chapter 4

Use Cases

4.1 Get gender prediction of journalistic source

The actor in the context of our project is a journalist or a newsroom who wants to be able to better understand the diversity of their sources. One of the diversity metrics newsrooms want to be better with is gender diversity. Having newsrooms target more women and non-binary people lead to better news, as a wider breadth of people are being heard from.

4.2 Get race prediction of journalistic source

Another one of the diversity metrics newsrooms want to improve is racial diversity. Much of the racial prejudice in this country is based on misinformation and unfamiliarity, but by having racial minorities' voices heard, many of the prejudices and biases people have may begin to disappear.

4.3 Help newsrooms add another layer of accuracy to the NLP prediction

Newsrooms want to make sure their predictions are as accurate as possible. The DEI toolkit has a certain level of accuracy which struggles in certain situations. We aim to mitigate the struggles of the DEI toolkit by adding another layer of accuracy so that the newsrooms can be more confident in their predictions.

4.4 Help newsrooms predict the race of common names

As previously stated, one of the current pitfalls of the DEI toolkit is that it struggles with common names like Austin Johnson, specifically names that can usually be either Black or White. With an image search, we can get a better idea of who the source is, and reduce the inaccuracy.

4.5 Help newsrooms predict the gender of gender-neutral names

Another one of the pitfalls of the current DEI toolkit is that it struggles with gender neutral names like Alex Morgan. With an image search, the system will be able to identify gender better than just by analyzing the name with natural language processing.

Chapter 5

Activity Diagrams and Risk Analysis

5.1 Activity Diagram

The activity Diagram below shows how our project will be used.

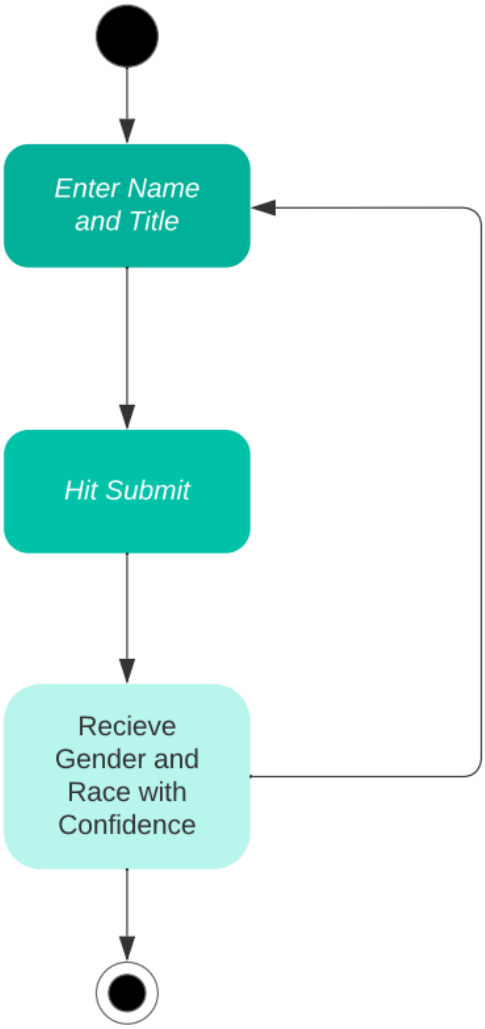

Figure 5.1: Activity Diagram

5.2 Risk Analysis

Risk	Consequences	Probability	Severity	Impact	Mitigation
Lack of knowledge	Delayed development	0.8	9	7.2	Doing sufficient research and experimenting with the technology
Project Scope	Less features than originally planned	0.6	7	4.2	Focus on critical requirements first
Bugs	Delayed development	0.9	5	4.5	Read documentation, review code

Figure 5.2 Risk Analysis

Our biggest roadblock this entire project has been our lack of knowledge. Going into this year with more background knowledge about how to setup a project like this would have saved us a significant amount of time, and we would have been able to expand our project scope to include more features

Chapter 6

Technologies Used

6.1 React

React is the frontend JavaScript-based framework.

6.2 Django

Django is the backend Python-based framework.

6.3 SerpAPI

SerpAPI is a free, third-party API (Application Programming Interface) used to procure image results from Google Images from a user query.

6.4 DeepFace API

6.4.1 Overview

DeepFace API is an open source API that can be used to analyze faces and return gender and ethnicity predictions along with confidence values

6.4.2 System Description

DeepFace utilized a hybrid model approach to find its prediction and confidence models. This means that multiple models were used and each was given a certain weight. Overall all of the models utilized the same four steps. These steps are detection, alignment, representation, and verification.

The detection phase is simply located where in a particular photo the face is. This is usually done through identifying features in the face like eyes, nose, mouth, or ears. The

face is sent to the alignment where the face is altered so that it is facing directly up and down and not at an angle.

The representation usually consists of a convolutional neural network which represents the face in a more digestible format for the computer. These neural networks will apply certain filters to the photos that emphasize certain features in the photo such as edges. They also break down the image into smaller files to increase run time.

The neural network's output is then sent to the verification stage where it is compared against a set of known photos. There are data sets for each of the ethnicities and genders and it is determined which groups the target photo is closest to. This is how the prediction is made and based on that similarity the confidence can be calculated.

6.5 GitHub

GitHub is a version control application used to track our software development.

Chapter 7

System Architecture

The original architecture of our system relied on two separate APIs to satisfy our requirements. The first API was Amazon Web Services (AWS) Rekognition which would generate the gender and confidence from a user query being saved in an AWS S3 Storage Bucket. From there, ClarifAI would return the ethnicity and confidence back to the API. While this worked, it wasn't ideal for two reasons. For one, it was quite slow waiting for two different APIs and having them analyze a face two separate times. Additionally, it led to a lot of problems if one API would fail, we would have an incomplete answer for the DEI toolkit. With those two problems, we decided to look elsewhere and that's where our final architecture comes.

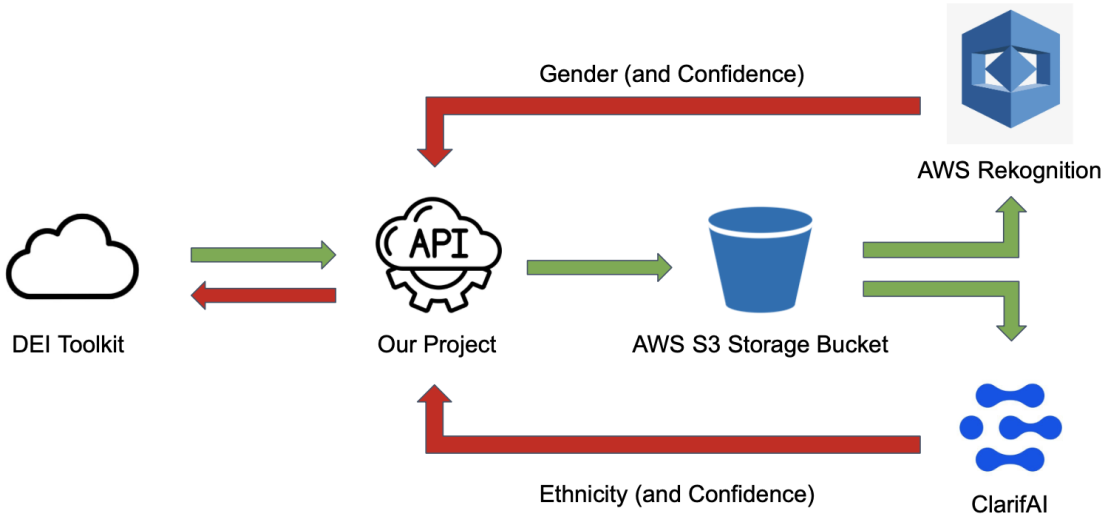

Figure 7.1: Original Architecture

The final system is built to be a centralized API (Application Programming Interface) that interacts with two other APIs: SerpAPI (represented by Google Image Search API) and DeepFace. Once our project receives a user query, we send a GET request to the Google Image Search API to get the first result from a Google Image search. From there, we send another GET request to the DeepFace API to get the race and gender predictions using that image as a parameter. Finally, our system would send those results back to the DEI Toolkit.

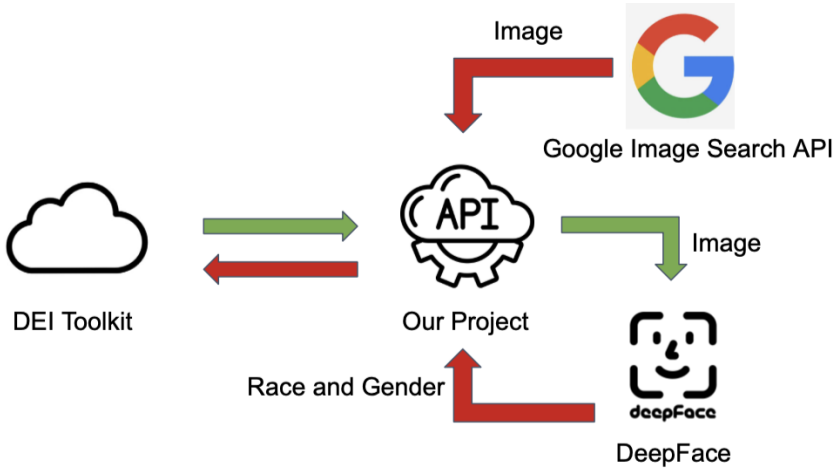

Figure 7.2: Final Architecture

Chapter 8

Design Rationale

8.1 User Interface

We decided to design our user interface with simplicity in mind for the goal of presenting the project during Senior Design. Originally, we planned to create an entirely backend service that would connect to the DEI Toolkit and abstract all the details of implementation. However, we realized having an interactive user interface that people could see would be a better way to showcase our work throughout the year. With that, our user interface is a simple textbox for a user to enter a query and click the submit button. From there, our user interface will show the picture that was obtained from the Google Search API along with the gender and ethnicity predictions from DeepFace.

8.2 Technology

We chose to use React for our frontend due to its simplicity. We were all familiar with React from other previous projects and felt it had all the specifications we were looking for. Additionally it was easy to connect with our Django backend which was a key aspect for this system.

Django was chosen as our backend for one reason: it was able to connect with DeepFace which is Python based. We originally built our backend using NodeJS and Express which are JavaScript based backend frameworks. While this worked, it was unable to connect with DeepFace and ultimately had to be rewritten for Django.

SerpAPI was chosen as it was an easy to use API for obtaining image results from Google Images. We originally thought of building our web-scraper to get these images; while this was an idea that would have worked, it was deemed to be too time consuming and thus using a pre-existing API was chosen. Additionally, since SerpAPI was free to use, it made the choice that much easier than other alternatives.

DeepFace was chosen as it was able to predict both gender and ethnicity based on computer vision. We looked at other face analysis APIs and none were able to give us both those characteristics. For example, major tech companies such as Google, Amazon, and Microsoft do not have ethnicity prediction capabilities due to ethical considerations. We started using Amazon's Rekognition service, but soon realized it was incomplete with our functional requirements.

GitHub was used for two reasons: our familiarity and its capabilities. All group members have used GitHub for other projects in the past and are familiar with its inner workings. Additionally, GitHub has great capabilities for tracking software development, testing, and code reviews: all important aspects for our Senior Design project and our pursuits of being Software Engineers.

Chapter 9

Test Plan

9.1 Unit Testing

We completed unit testing throughout our development testing each modular aspect of our code. Once a module was created, such as integrating SerpAPI, we would create unit tests to ensure each functional requirement worked properly. This usually would involve mocking different services to ensure each service worked independently and could be integrated into the entire system without faults.

9.2 Integration Testing

Unfortunately, we had to adjust priorities of our Senior Design project due to time constraints from learning new technologies and the persistent bugs. This adjustment led to focusing on creating a solid API that would eventually be integrated into the DEI Toolkit. This future work would be straightforward due to our good software engineering principles we implemented with this system.

Chapter 10

Description of System Implementation

Once the system is booted up, users are presented with a blank textbox and a submit button for queries. Once the user enters a query and clicks submit, the submit button will become disabled thus letting the user know our system is working to find the image of the desired query and analyzing it. Once the analysis is done, the user will be presented with the text of their query, image, and gender and ethnicity predictions. The user could then enter another query and the process would repeat.

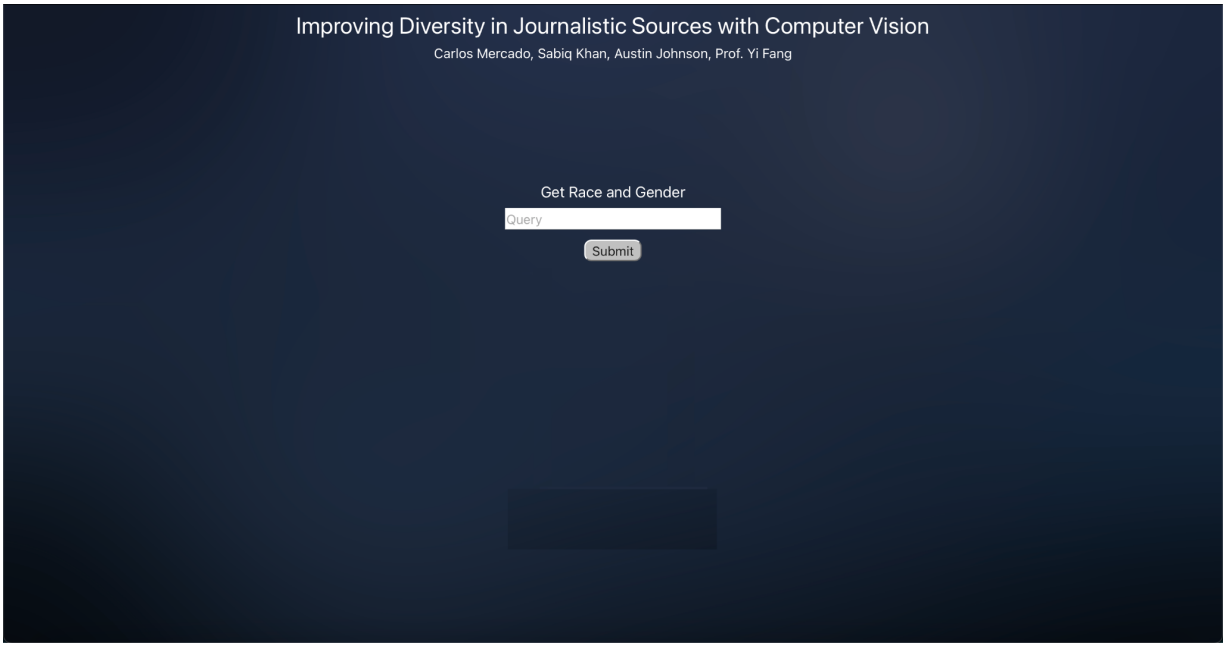

Figure 10.1: Welcome Screen

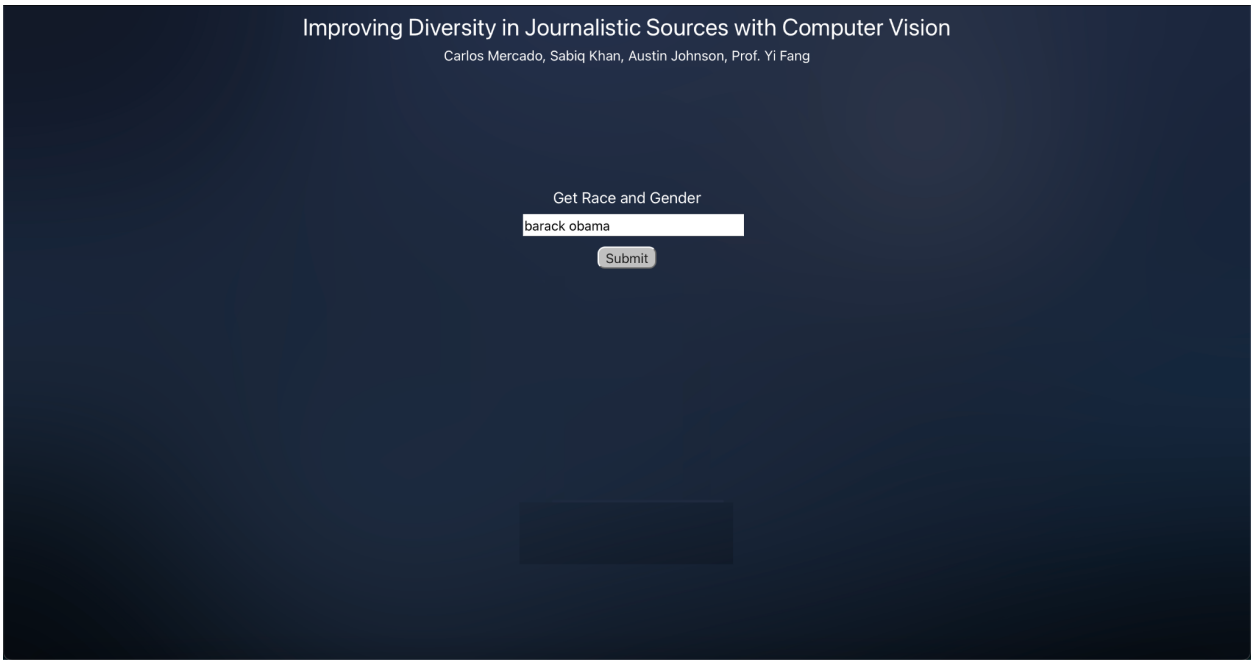

Figure 10.2: User enters query

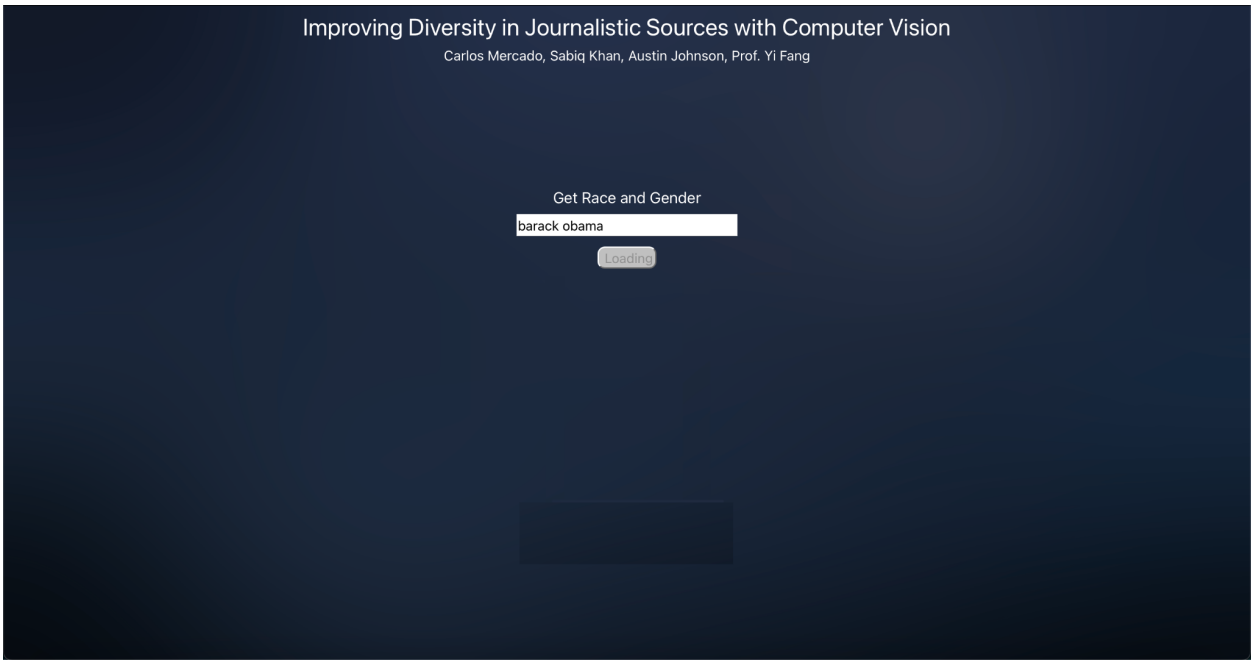

Figure 10.3: User submits query and button becomes disabled and says “Loading”

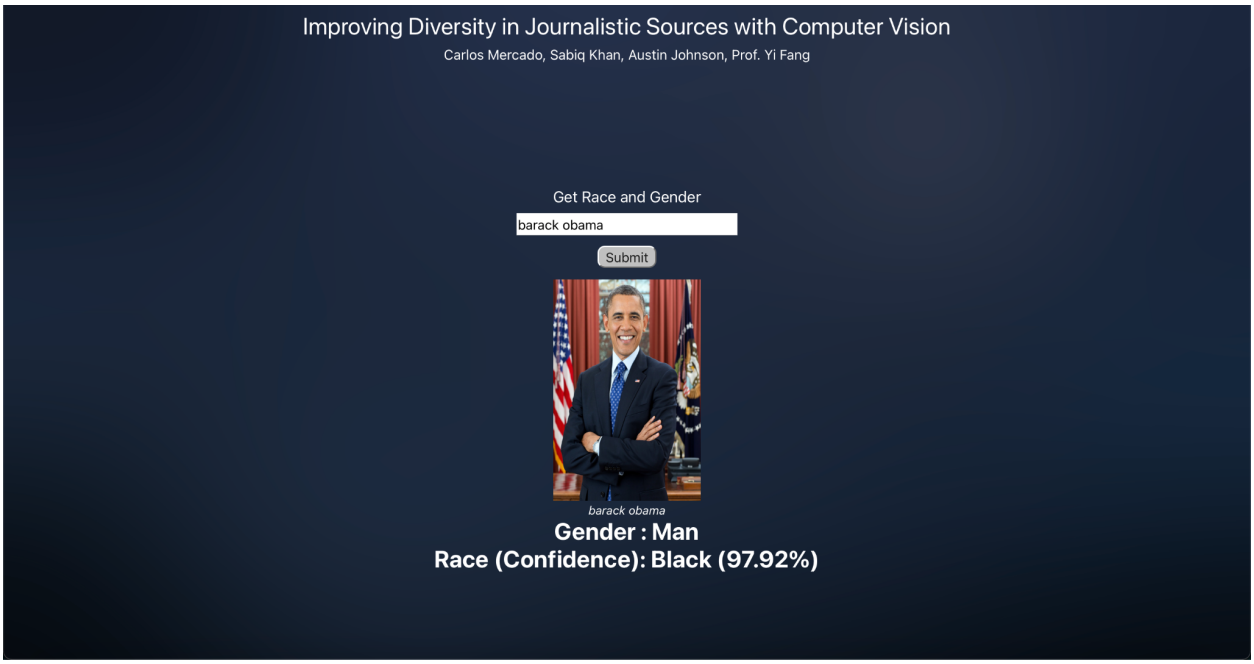

Figure 10.4: User query, image, gender and ethnicity predictions are shown

Chapter 11

Social Implications

11.1 Ethical

The Ethical ramifications of our project were a prominent part of discussions when deciding what we should do and how we should move forward. The use of facial recognition software, and specifically racial detection with facial recognition has been met with criticism and mistrust in recent years. One thing we noticed while attempting to find the most effective facial recognition software was that major machine tech companies (eg. Microsoft, Google, Amazon, Facebook) no longer provided race detection in their facial recognition software. Many of the companies had made statements about how they originally had the feature, but eventually removed it, as a result of backlash, or ethical introspection. This was the first major red flag for us, as our project was rooted in race detection from facial recognition.

After further research, we discovered and learned more about how facial recognition had been used to discriminate against a multitude of ethnic groups and minorities.

As a group we understand how race detection in facial recognition software can be used. It is employed for law enforcement surveillance, airport passenger screening, and employment and housing decisions. Despite widespread adoption, face recognition was recently banned for use by police and local agencies in several cities, including Boston and San Francisco. Why? Of the dominant biometrics in use (fingerprint, iris, palm, voice, and face), face recognition is the least accurate and is rife with privacy concerns.

We acknowledge that our project implements facial recognition and race detection, but our end goal is to foster diversity, not to foster discrimination. This is why we felt it was appropriate to use racial detection software. The tool may be used for unjust causes, but the tool itself is not inherently unjust.

11.2 Social

Through this project, we aim to help minorities and marginalized groups become more heard. Newsrooms will be able to pinpoint exactly where they are primarily getting their sources, and as a result, will be able to tackle social issues facing these marginalized and minority groups, not usually heard from.

11.3 Political

Another large impact our project could have is to tackle the misrepresentation of political views of many of these minority and marginalized groups. By hearing more about their stories, people may shift their own political views.

11.4 Economic

This project is available for any newsroom, big or small, to use, so the economic ramifications in regards to the newsroom are minimal. In regards to society, many of the socio economic issues facing our population will be more likely to come to light and be more evident as a result of our project.

11.5 Health and Safety

Our project is easy and safe to use. There are no health or safety implications of using this project on your device. Any risk would be of simply using the device.

11.6 Manufacturability

This project follows industry standard technology, and is simple to install, use, and maintain.

11.7 Sustainability

This project will remain useful while journalists source their news. Journalists moving forward will be more cognizant of their sources and where they’re coming from.

11.8 Environmental Impact

The environmental impact of our app is no greater than the already existing environmental impact of running the devices journalists use, so there is a slim environmental impact.

11.9 Usability

Our project is very simple and easy to use. The functionality is intuitive and follows standard UI/UX design principles so that the user can apply previous intuition and knowledge to easily make progress.

Chapter 12

Difficulties Encountered and Lessons Learned

12.1 Difficulties

Throughout this project, we faced many different difficulties. First was determining the scope of our project. For the year, our initial goal was to create our system and to also implement our system with the existing DEI toolkit. After consultation with our advisor, we deemed that was out of our scope and that we should focus primarily on the construction and completion of our own system.

Our next significant road-bump was testing different facial recognition technologies. For V1 of our project, we used Amazon Rekognition facial recognition software for our gender detection and a ClarifAI for the race detection. This was difficult because using two APIs at once, along with making sure both worked at the same time, was inconsistent. We also tested many different APIs, but many of them either were difficult to implement, or were much less accurate.

SerpAPI was also a difficulty because there were not an unlimited amount of free uses. We were constrained to 100 free google searches, so our testing capabilities were significantly hamstrung, as we were only able to test 100 times total throughout the year.

Finding the right UI was also a minor difficulty, as many of the different ideas we had were either too busy, too simple, or difficult to look at for a long period of time.

12.2 Lessons Learned

Some of the lessons we learned throughout this design process were the necessity to communicate openly and freely with our team, make sure to set goals and deadlines, along with sticking to said goals and deadlines, and making outlines of what we need to code, and taking a modular approach to it.

In terms of communication, we found we were able to more effectively meet deadlines and goals when we were texting frequently and keeping in touch about what we were currently doing, and what needed to be done soon.

In terms of setting deadlines, we had many weeks where there were no set deadlines, and general progress was being made much slower than usual. Once we set deadlines for ourselves, regardless of advisor, we were able to more effectively and efficiently finish our work.

And finally, in terms of outlining what we need to code, computer science projects are often difficult to split up. Much of it can be done by one person who has the best expertise if they write one large monolithic file of code. But we realized the best way to move forward with working on this project was to split up the project into modules, so that we could work on it together most efficiently.

Chapter 13

Future Work

Moving forward, we would like to merge the system we created with the DEI toolkit. In practice, there is no UI for our project, but instead simply a backend, where we would receive a name from the DEI toolkit, and process it and return a result, like our project does currently. But instead of displaying the result on a page, we would simply return the result back to the DEI toolkit where it could store the data in a database and display the data to its users.

We would also like to potentially add age as a diversity metric. Age is something that cannot be determined simply from a name, and an image can do well to predict one's age. Many of these facial detection technologies predict the ages of the image successfully and consistently

Appendix A

Installation Guide

A.1 Getting code from GitHub

Clone the repository and change to that directory in your local environment

```
git clone https://github.com/carlosm11j/seniordesign.git
cd seniordesign
```

A.2 Installing Dependencies

Next you will have to install all the necessary dependencies for Django to run properly

```
pip install google-search-results
pip install deepface
```

A.3 Running System

Finally, we will start both the frontend and the backend

```
cd frontend
npm start
```

Open a new terminal window and run

```
cd env/seniordesign
python manage.py runserver
```

Bibliography

[1] DeepFace API. https://github.com/serengil/deepface

[2] AWS Rekognition. https://aws.amazon.com/rekognition/

[3] Django Framework. https://www.djangoproject.com

[4] SerpAPI: https://serpapi.com

[5] Tal Abbady: The Modern Newsroom Is Stuck Behind The Gender And Color Line, May 2017. https://www.npr.org/sections/codeswitch/2017/05/01/492982066/the-modern-newsroom-is-stuck-behind-the-gender-and-color-line

[6] Jeanne Bourgault: Diversity in the newsroom can build better media. Here's why, Dec 2021. https://www.weforum.org/agenda/2021/12/diversity-in-news-media/

[7] Xiaoxiao Shang, Zhiyuan Peng, Qiming Yuan, Sabiq Khan, Lauren Xie, Yi Fang and Subramaniam Vincent. DIANES: A DEI Audit Toolkit for News Sources. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.

Improving_Diversity_In_Journalistic_Sources_With_Computer_Vision_Publication

Final Audit Report 2022-06-13

Created:	2022-06-10
By:	Darcy Yaley (dyaley@scu.edu)
Status:	Signed
Transaction ID:	CBJCHBCAABAAcHQ8L6aEsqdy-c66fMJEX0ZuirEj1OXM

"Improving_Diversity_In_Journalistic_Sources_With_Computer_Vision_Publication" History

Document created by Darcy Yaley (dyaley@scu.edu)
2022-06-10 - 4:36:57 PM GMT

Document emailed to Yi Fang (yfang@scu.edu) for signature
2022-06-10 - 4:38:03 PM GMT

Email viewed by Yi Fang (yfang@scu.edu)
2022-06-10 - 4:38:46 PM GMT

Document e-signed by Yi Fang (yfang@scu.edu)
Signature Date: 2022-06-13 - 8:53:48 PM GMT - Time Source: server

Document emailed to N. Ling (nling@scu.edu) for signature
2022-06-13 - 8:53:50 PM GMT

Email viewed by N. Ling (nling@scu.edu)
2022-06-13 - 8:53:53 PM GMT

Document e-signed by N. Ling (nling@scu.edu)
Signature Date: 2022-06-13 - 9:29:00 PM GMT - Time Source: server

Agreement completed.
2022-06-13 - 9:29:00 PM GMT