

AN INTELLIGENT VIDEO EDITING AUTOMATE FRAMEWORK USING AI AND COMPUTER VISION

Haolin Xie¹ and Yu Sun²

¹Northwood High School, 4515 Portola Pkwy, Irvine, CA 92620

²California State Polytechnic University, Pomona,
CA, 91768, Irvine, CA 92620

ABSTRACT

At present, many video editing software have been created, but what they all have in common is that they require manual work to edit. And it takes a lot of time and the user needs to watch each frame before editing. In this paper, we have developed a program about AI intelligence. The most important point of this software is that it can automatically focus the face of a person and edit only selected clips of the person to make a complete video. Users only need to prepare the video they want to edit and a photo of the main character. Then, upload both to the software and AI Intelligence will automatically edit it, providing the user with a way to download and save it after editing the main character they need. we applied our application to an example video using the Marvel character Hawkeye as he appears in End Game, and tried many experiments with the clip, eventually we tried many experiments with the clip and finally got a video of our selected character. The results show that this software saves the user a lot of time and is highly efficient. All operations are carried out by AI.

KEYWORDS

Software, API, Face recognition.

1. INTRODUCTION

At present, many video editing software has been created, but what they all have in common is that they require manual work to edit. And it takes a lot of time and the user needs to watch each frame before editing. In this paper, we have developed a program about AI intelligence. The most important point of this software is that it can automatically focus on the face of a person and edit only selected clips of the person to make a complete video. Users only need to prepare the video they want to edit and a photo of the main character. Then, upload both to the software and AI Intelligence will automatically edit it, providing the user with a way to download and save it after editing the main character they need. The results show that this software saves the user a lot of time and is highly efficient AI carries out all operations. The purpose of the theme of this software is to provide the user with the opportunity to edit a video of a specific person of their choice, and the edited video will only feature the person chosen by the user.

There is no software or web page similar to our software available to users. But there are many methods and systems that can perform face recognition. Some of the face detection techniques and systems that have been proposed to improve surveillance and help track criminals and terrorists, allow the user to enhance personal security and also extend to convenience. For example, many Apple products are available with Face ID, which eliminates the need for users to

David C. Wyld et al. (Eds): ICAIT, CBIoT, WiMo, CRYPIS, ICDIPV, CAIML, NLCA - 2022
pp. 201-209, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121216

enter a password, and also enhances user privacy and security. MediaPipe provides an ultra-fast face detection solution with six landmarks and multi-face support [3][4]. It is based on BlazeFace, a lightweight and well-performing face detector designed for mobile GPU inference. Facial recognition technology doesn't always work as well as it should [1]. Facial recognition systems can be affected by poor lighting or low image quality, which can prevent accurate recognition, and this improves the computer's algorithm and error rate [2][5]. Or the data may not match the person's nodes because the camera angle is obscured; this creates errors when matching facial prints cannot be verified in the database. Because all faces are physically similar to each other, each person's facial parts are in the exact location [6].

In this paper, we follow the same line of research by face recognition. Our goal is to improve the accuracy and degrade the error rate of face recognition and provide the user with a video with only the parts that the user needs, and do so with as little error rate as possible to save the user's time and improve productivity. First, identify and locate the facial features to get the coordinates of the eyes, ears, cheeks, nose, and mouth of each detected face. And get the contours of the detected face and its eyes, eyebrows, lips, and nose. Third, the images and videos are grayed out to avoid overexposure or other elements that cause lighting problems. The whole system will not affect the success of the final result after ensuring that the lighting in the original image is uniform and that the facial features can be accurately detected. This is also a great help to our software, as we believe that the system can accurately identify the people in the video and edit the video.

We use the Face Detection API that is now available to help our software a lot and apply it to our software. We practice two different methods to detect the success of the software. First, we used the Marvel character Hawkeye and the Avengers: Endgame trailer for this test. After we uploaded both of them, the video generated by the software system after editing was Hawkeye, and after several attempts, the system generated the same video as the previous ones. We then used the same photo to test all the videos of Hawkeye in Marvel, and our software was able to generate the same video with only Hawkeye. Secondly, we used different characters from the Endgame trailer to test the same software system to edit the video successfully, and there were no errors, which shows the usefulness of our method and system.

The rest of the paper is organized as follows: Section 2 gives the details on the challenges that we met during the experiment and designing the sample; Section 3 focuses on the details of our solutions corresponding to the challenges that we mentioned in Section 2; Section 4 presents the relevant details about the experiment we did, following by presenting the related work in Section 5. Finally, Section 6 gives the concluding remarks, as well as points out the future work of this project.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Making the Software Easier to Use

When I chose the software to make this theme, I was confused because I would struggle with what to add to make the software look better or easier to use. When designing the interface of the software, I struggled with the layout, such as where to place the icons for uploading photos and videos. I wasn't sure how to use the GUI and the information it required to exchange between software and the amount of memory and processing power they needed to use. The challenge is to use the GUI to make the App look aesthetically better and organize its features.

2.2. Passing Every Frame

After executing the Face Detection API for face recognition, the next challenge is to pass every frame and every frame in which the person appears to the next code for the operation. And how to stitch together the images of each appearance of the character, and then need to avoid is the character after a flash and then no picture, so also need to edit out the character appeared before and after each video left a second or two, so that the character will not be a flash and thus can not see the appearance of the character, these are the challenges and difficulties to face.

2.3. Developing Software into Android and IOS

After the system and AI part of the software is solved, it needs to be developed into Android and IOS software [7]. And maintenance and building are the issues that need to be faced. The development of Android and IOS applications involves a lot of complexity. For example coding multiple files for one screen, problem-solving and bug fixing, and compatibility with all Android devices including IOS as well [8]. Each step is a new challenge and after solving one problem another new one may appear, so this is the biggest difficulty and one of the most time-consuming parts of developing this software.

3. SOLUTION

We needed to solve the problem of how to use the GUI to make the application look more beautiful and organize its functions, how to stitch together the images of the characters each time they appear, and then we needed to avoid having no images after the characters flicker and we needed to develop the software for Android and IOS and make sure that most of the IOS and Android devices are compatible [9]. We want to save more time for users by providing software for users to edit their videos to improve efficiency. And we want to demonstrate through this research that face recognition is widely available and can be integrated into people's daily lives, just like our software, which uses face recognition to edit a complete video.

Python GUI project is a simple API for developers to create user interfaces using native elements of Python applications [10]. As a lightweight API, not much code is required between the application and the target platform, which makes it more efficient than many other frameworks on this list. Our software needs to use the GUI to make the application look good and to lay out our interface. Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. and the open-source community.

```
_loadHistory() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  List dates = [];
  print(prefs.containsKey('date'));
  if (prefs.containsKey('date')) {
    dates = prefs.getStringList('date')!;
    for (var date in dates) {
      _history[date] = json.decode(prefs.getString(date)!);
    }
  }

  print(_history.keys);
  setState(() {});
}
```

Figure 1. Code of how history is stored

Here is an example of code I wrote for my project: The code on the right shows how the video history is accessed/stored.

```

Future<void> _setVideoController() async {
  File file = File(widget.videoFilePath);
  if (file != null && mounted) {
    VideoPlayerController controller;
    print('play video ');
    if (kIsWeb) {
      controller = VideoPlayerController.network(file.path);
      print('network:' + file.path);
    } else {
      controller = VideoPlayerController.file(File(file.path));
      print('file:' + file.path);
    }
    setState(() {
      _controller = controller;
    });
  }
}

```

Figure 2. Code of loading video information

The code on the right shows the class in charge of loading video information to be edited by the programmed AI.

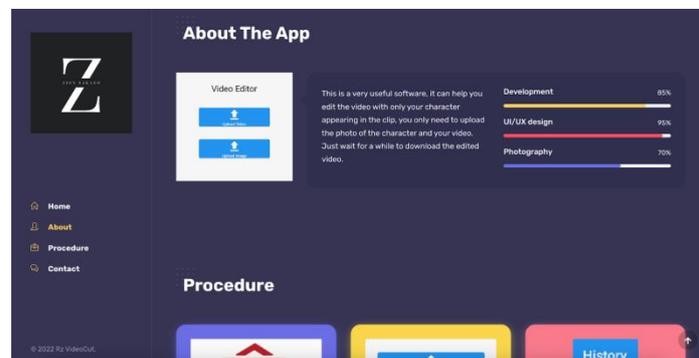


Figure 3. Screenshot of the App

```

173 <section id="about">
174
175 <div class="container">
176
177 <!-- section title -->
178 <h2 class="section-title wow fadeUp">About The App</h2>
179
180 <div class="spacer" data-height="80"></div>
181
182 <div class="row">
183
184 <div class="col-md-3">
185 <div class="text-center text-md-left">
186 <!-- avatar image -->
187 
188 </div>
189 <div class="spacer d-md-none d-lg-none" data-height="38"></div>
190 </div>
191
192 <div class="col-md-9 triangle-left-md triangle-top-sm">
193 <div class="rounded bg-dark shadow-light padding-30">
194 <div class="row">
195 <div class="col-md-6">
196 <!-- about text -->
197 <p>This is a very useful software, it can help you edit the video with only your character appo-
198 <div class="mt-3">
199 <!-- social icons -->
200 <div class="row">
201 <div class="spacer d-md-none d-lg-none" data-height="38"></div>
202 </div>
203 <div class="col-md-6">
204 <!-- skill item -->
205 <div class="skill-item">

```

Figure 4. Screenshot of code

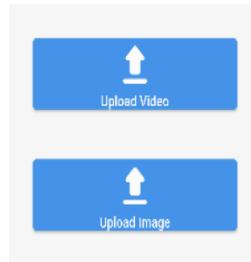


Figure 5. Screenshot of upload page

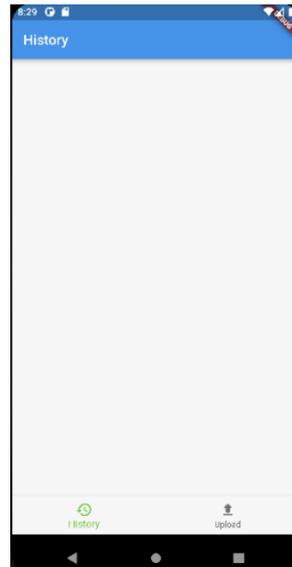


Figure 6. Screenshot of history page

There are many free panels or templates available on the web for the general public, and it is easy to reposition or re-color each button using the templates; there is not much code required between the GUI application and the target platform. This makes it more efficient than other programming languages or methods of creating frameworks, and it is good for solving the aesthetics of our software and organizing its functionality. Then the face recognition will scan one side of the video uploaded by the user, edit it and send it back to the port so that the user can download and save the edited video. AWS is a widely adopted cloud platform that provides a variety of on-demand operations such as computing power, database storage, content delivery, etc. This step solves the problem of receiving videos as well as photos uploaded by users and then proceeding to the next step. Developing software for IOS and Android both require different programming languages. development for IOS can be more cumbersome and only requires improving the readability of constants. The best way to use structures in Swift is to create a file for all constants in the application.

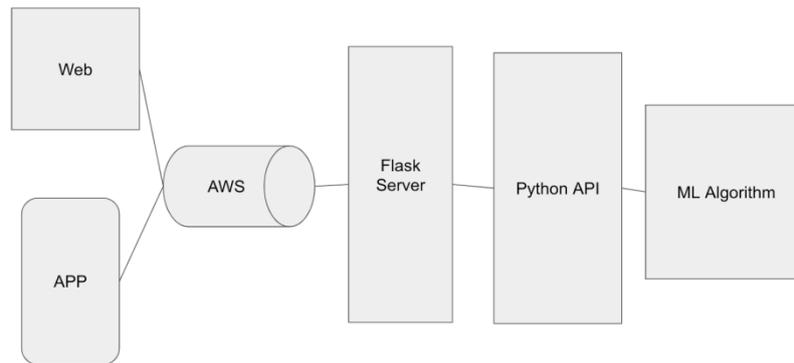


Figure 7. Overview of the system

The development of this software uses a GUI for aesthetic interface and functionality, AWS for computation and database storage, APIs to use it with any other programming language, and ML algorithms to allow the machine to follow a set of instructions to perform a task; the algorithms also help the machine to choose and decide which set of instructions will produce better results.

Mr. Evan Gunnell helped me and guided me together to solve the problem of getting the frame after face recognition and each frame and composing it into a video, as well as solving the problem that there is no more video when the character flashes by. And Mr. Armando helped me with the aesthetics and layout of the software interface and developing it for Android and Apple. We use the same character and test it several times on the same video to see if the edited video is the same and alive with errors. Similarly, using different characters in the same video for multiple tests, the clips came out with the corresponding characters.

It ensures the correctness of the software and the multifaceted connection of the tools to make the whole software work properly. We use the current face detection to help our software a lot and apply it to our software.

There are many methods as well as tips available on the internet when it comes to problems, but sometimes these may not be useful and we may need to incorporate all the tips and methods in order to function.

4. EXPERIMENT

4.1. Experiment 1

My solution was to use the same person in the same video for 5-8 tests and also use different photos of the person but use the same video for multiple tests. This way we can test the correctness of this software as well as the error rate. First, we prepared the trailers of Hawkeye and Avengers: Endgame with Marvel characters, and uploaded them to the software. AWS will start the operation to store the photos and videos and then do the face recognition and edit them to download them and repeat the above steps to ensure the accuracy rate. The final result is that the edited videos are identical and no other characters appear inside.

4.2. Experiment 2

My second solution was to use the same character but change different photos but still use the same video for multiple tests. This way I could also test the correctness and error rate of the

software [11]. The other solution is to use different characters but the same video for the test. First, again, we prepare many photos of Hawkeye and upload them to the software for testing, use each photo to edit the video, and then repeat the test with different photos. After several tests, the edited video was the same as the other edited video and there was no difference, so the experiment was successful.

The results of the experiment are up to expectations because after uploading new photos and videos, the face recognition will re-identify one side of the photos and videos instead of existing in one data, it will not overlap with the previous data, so it will not lead to the next editing of the video and face recognition and thus will not affect the results of the test. The results of the test show that the face recognition and editing of the composite video are successful [12]. We tried many different factors, but the software achieved the desired goal.

5. RELATED WORK

Limin Wang and his team study appearance and relational networks for video classification, focusing on learning video representation in an end-to-end manner [13]. Their team also focuses on the goal of each frame, but unlike our software, they connect the appearance for spatial modeling. And their team has gained significant improvements over 3D convolution in Spatio-temporal feature learning.

S Janhavi and Chandra Sekhar Malepati use video classification for real-time human activity recognition, which is very similar to our project because we need to edit a video after face recognition [14]. And theirs is the ability to recognize the activity of others, and again, both are in a moving state. I think their project is interesting because it can detect what kind of movement people are doing, which will help in the future.

Paul Viola & Michael J. Jones has created powerful real-time face detection, which is similar to my software project, and also sports face recognition [15]. But the difference is that their project is real-time while my software requires user uploads. And they used the AdaBoost learning algorithm (Freund and Schapire, 1995) to build a simple and efficient classifier for selecting a small number of key visual features from a large number of potential features, which greatly improves the speed of face detection and the accuracy of real-time detection.

6. CONCLUSIONS

This software is a program about AI intelligence. The most important point of this software is that it can automatically focus on a person's face and edit only selected clips of that person to create a complete video. Users only need to prepare the video they want to edit and a photo of the main character. Then, both are uploaded to the software and AI Intelligence automatically edits them, providing a way for users to download and save them after editing the desired protagonist. And AWS was used to store the data, with the goal of providing users with a way to upload as well as download the edited video. The software was tested repeatedly to ensure accuracy and downgrade the error rate. The results show that the software saves users a lot of time and is efficient, with all operations performed by artificial intelligence.

This software is a program about AI intelligence. The most important point of this software is that it can automatically focus on a person's face and edit only selected clips of that person to create a complete video. Users only need to prepare the video they want to edit and a photo of the main character. Then, both are uploaded to the software and AI Intelligence automatically edits them, providing a way for users to download and save them after editing the desired protagonist.

And AWS was used to store the data, with the goal of providing users with a way to upload as well as download the edited video. The software was tested repeatedly to ensure accuracy and downgrade the error rate. If there is more time, I will improve the interface and content of the software and add more information for the And adding details makes the interior of the software look more concise. If I were to continue to work on this project, I would add these three things: 1. add subtitles. 2. Improve the GUI. 3 .Background Music.

And this software is currently available for download on the Google Play Store, link: https://play.google.com/store/apps/details?id=com.codingminds.video_editor_flutter

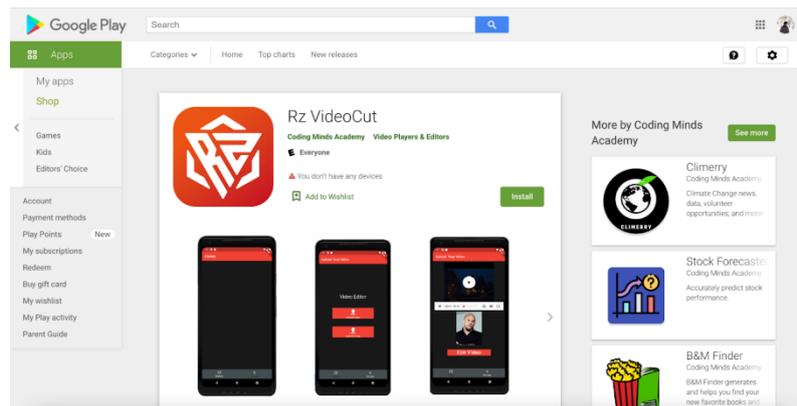


Figure 8. Screenshot of App

REFERENCES

- [1] Kaur, Paramjit, et al. "Facial-recognition algorithms: A literature review." *Medicine, Science and the Law* 60.2 (2020): 131-139.
- [2] Gorodnichy, Dmitry O. "Facial recognition in video." *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, Berlin, Heidelberg, 2003.
- [3] Lugaresi, Camillo, et al. "Mediapipe: A framework for perceiving and processing reality." *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)*. Vol. 2019. 2019.
- [4] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." *arXiv preprint arXiv:1906.08172* (2019).
- [5] Duchaine, B. "Individual differences in face recognition ability: Impacts on law enforcement, criminal justice and national security." *Psychological Science Agenda* (2015).
- [6] Liu, Haowei. "Face technologies on mobile devices." *Facial Detection and Recognition on Mobile Devices* (2015): 11-38.
- [7] Enck, William, et al. "A study of android application security." *USENIX security symposium*. Vol. 2. No. 2. 2011.
- [8] Chin, Erika, et al. "Analyzing inter-application communication in Android." *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 2011.
- [9] Liu, Jianye, and Jiankun Yu. "Research on development of android applications." *2011 4th International Conference on Intelligent Networks and Intelligent Systems*. IEEE, 2011.
- [10] Podrzaj, Primoz. "A brief demonstration of some Python GUI libraries." *Proceedings of the 8th International Conference on Informatics and Applications ICIA2019*. 2019.
- [11] Madhuri, T., and P. Sowjanya. "Microsoft Azure v/s Amazon AWS cloud services: A comparative study." *International Journal of Innovative Research in Science, Engineering and Technology* 5.3 (2016): 3904-3907.
- [12] Real, Esteban, et al. "Automl-zero: Evolving machine learning algorithms from scratch." *International Conference on Machine Learning*. PMLR, 2020.

- [13] Wang, Limin, et al. "Appearance-and-relation networks for video classification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [14] Janhavi, S., and Chandra Sekhar Malepati. Real Time Human Activity Recognition with Video Classification. No. 7377. EasyChair, 2022.
- [15] Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2004): 137-154.

© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.