

Table of Contents

Table of Contents	1
List of Figures	
1.CHAPTER	2
INTRODUCTION	2
1.1Explain Screen Recording	2
1.2History & Evolution of Screen Recording Technology	2
2.CHAPTER	3
WHAT IS SCREEN RECORDING?	3
2.1An Overview of Screen Recording Sowftware and Their Features	3
2.2Latest Trends and Innovations in Screen Recording Technology	4
3.CHAPTER	5
 Requirement Analysis	5
3.1 Software Requirements	5
4.CHAPTER	6
 How It Works	6
4.1 Implementation	6
RESULTS & DISCUSSION	10
CONCLUSION	11
APPENDIX	12
BIBLIOGRAPHY	15

1. CHAPTER

INTRODUCTION

1.1 Explain Screen Recording

In recent years, the use of screen recording technology has exploded, with a growing number of individuals and organizations turning to this technology as a way to capture and share information and ideas. From software development and testing to online education and training, screen recording technology has become an indispensable tool for those seeking to communicate and collaborate more effectively in today's digital world. This research paper will examine the evolution of screen recording technology, its current state, and its future potential, including a discussion of the various tools and platforms that are available, their features and capabilities, and the benefits and limitations of this technology. This research paper will explore the growth and development of screen recording technology, including a discussion of its uses in fields such as software development, online education, and instructional design. Today, there are many different types of screen recording software and hardware available, ranging from free and opensource tools to high-end, professional-level solutions. Additionally, this paper will examine the various tools and platforms that are available, their features and capabilities, and the benefits and limitations of screen recording technology.

1.2 History & Evolution of Screen Recording Technology

Screen recording technology has been around for several decades, but it has undergone significant improvements and advancements over time. The earliest screen recording software dates back to the 1990s, but it was primarily used for creating software demonstrations and training videos. In the early 2000s, with the rise of online video platforms like YouTube and Vimeo, screen recording became more popular as a way to create tutorials and instructional content. In 2004, the first version of Camtasia was released, which became a popular screen recording tool for both Windows and Mac.

2. CHAPTER

WHAT IS SCREEN RECORDING?

Screen recorder is software that captures content and activities that take place on a computer screen. Screen recorder is useful during tasks such as creating video tutorials, recording screen content, etc.

2.1 An Overview of Screen Recording Software and Their Features There are several types of screen recording software available, each with its own set of features.

Here are some of the most common types:

- **Built-in screen recording:** Some operating systems have built-in screen recordings functionality, such as Windows 10 Game Bar or macOS QuickTime Player.
- **Standalone software:** These are programs that are specifically designed for screen recordings, such as Camtasia, Snagit, or OBS Studio. They often offer more advanced features, such as editing tools, webcam recording, or live streaming options.
- **Web-based screen recorders:** These are online tools that allow you to record your screen without downloading any software. Examples include Screencast-O-Matic or Loom
- **Mobile screen recorders:** These are apps that allow you to record your mobile device's screen, such as AZ Screen Recorder or DU Recorder.

Features vary depending on the type of software, but common features include recording video and audio, selecting a specific area of the screen to record, adding text or annotations, adjusting video quality and frame rate, and exporting videos in different formats. More advanced features may include green screen effects, multitrack editing, and integration with other software tools.

2.2 Latest Trends and Innovations in Screen Recording Technology

The current screen recording technology includes the availability of software applications that enable users to capture and record their computer screens in high resolution with minimal lag. The latest trends and innovations in screen recording technology include the use of artificial intelligence to automate the video editing process and the integration of cloud-based storage options to enable easy sharing and collaboration. Additionally, new screen recording tools are being developed that provide features such as real-time collaboration, live streaming, and video annotation. Overall, the screen recording technology industry is rapidly evolving, with new innovations being introduced regularly to enhance user experience and increase functionality.

Some specific trends and innovations in screen recording technology include

- Cloud-based storage and sharing: Screen recording software now often includes the option to save recordings directly to the cloud, making it easy to share videos with others and access them from multiple devices.
- Artificial intelligence (AI) editing: Some screen recording tools now use AI algorithms to automatically edit recordings, removing pauses and filler words, and adding captions or animations.
- Real-time collaboration: New screen recording software is being developed that allows multiple users to collaborate on a single recording in real time, providing a powerful tool for remote teams to work together on projects.
- Live streaming: Screen recording technology is increasingly being used for live streaming, allowing users to broadcast their computer screen in real time to an audience.
- Video annotation: Screen recording software is being developed that includes tools for adding annotations, such as arrows or text, to the recording to provide context or highlight important information.
- Integration with video conferencing software: Some screen recording software now integrates with video conferencing tools, allowing users to record their video meetings for later reference.

Overall, these trends and innovations are making screen recording technology more powerful and versatile, with new features being added all the time to enhance user experience and make it easier to create high-quality videos.

3. CHAPTER

REQUIREMENT ANALYSIS

3.1 Software Requirements

- Operating System : Windows 7 or above
- Programming Language : Python
- Python Libraries : Numpy ,Pyautogui,Opencv-python
- Python IDE : VS Code , Jupyter ,PyCharm etc.

4. CHAPTER

How It Works

4.1 Implementation

Importing the required packages:

- Import Pyautogui
- Import Cv2
- Import Numpy

```
import pyautogui
import cv2
import numpy as np
```

Fig.1: Import Library

Now, before recording the screen, we have to create a VideoWriter object. Also, we have to specify the output file name, Video codec, FPS, and video resolution. In video codec, we have to specify a 4-byte code (such as XVID, MJPG, X264, etc.). We'll be using XVID here.

```
# Specify resolution
```

```
resolution = (1920, 1080)
```

```
# Specify video codec
```

```
codec = cv2.VideoWriter_fourcc(*"XVID")
```

```
# Specify name of Output file filename
```

```
= "Recording.avi"
```

```
# Specify frames rate. We can choose
```

```
# any value and experiment with it fps
```

```
= 60.0
```

```
# Creating a VideoWriter object
```

```
out = cv2.VideoWriter(filename, codec, fps, resolution)
```

```

# Specify resolution
resolution = (1920, 1080)

# Specify video codec
codec = cv2.VideoWriter_fourcc(*"XVID")

# Specify name of Output file
filename = "Recording.avi"

# Specify frames rate. We can choose
# any value and experiment with it
fps = 60.0

# Creating a VideoWriter object
out = cv2.VideoWriter(filename, codec, fps, resolution)

```

Fig.2: Creating Video Editor

To display the recording in real-time, we have to create an Empty window and resize it.

Create an Empty window

```
cv2.namedWindow("Live", cv2.WINDOW_NORMAL)
```

Resize this window cv2.resizeWindow("Live",
480, 270)

```

# Create an Empty window
cv2.namedWindow("Live", cv2.WINDOW_NORMAL)

# Resize this window
cv2.resizeWindow("Live", 480, 270)

```

Fig.3:Creating Empty Window

Now, let's start recording our screen. We will be running an infinite loop and in each iteration of the loop, we will take a screenshot and write it to the output file with the help of the video writer. while True:

```
# Take screenshot using PyAutoGUI
img = pyautogui.screenshot()

# Convert the screenshot to a numpy array
frame = np.array(img)

# Convert it from BGR(Blue, Green, Red) to
# RGB(Red, Green, Blue)          frame =
cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Write it to the output file
out.write(frame)

# Optional: Display the recording screen
cv2.imshow('Live', frame)

# Stop recording when we press 'q'
if cv2.waitKey(1) == ord('q'):
    break
```

```

while True:

    # Take screenshot using PyAutoGUI
    img = pyautogui.screenshot()

    # Convert the screenshot to a numpy array
    frame = np.array(img)

    # Convert it from BGR(Blue, Green, Red) to
    # RGB(Red, Green, Blue)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Write it to the output file
    out.write(frame)

    # Optional: Display the recording screen
    cv2.imshow('Live', frame)

    # Stop recording when we press 'q'
    if cv2.waitKey(1) == ord('q'):
        break

```

Fig.4:Write The Output File

After everything is done, we will release the writer and destroy all windows opened by OpenCV.

Release the Video writer out.release()

Destroy all windows

cv2.destroyAllWindows()

```

# Release the Video writer
out.release()
# Destroy all windows
cv2.destroyAllWindows()

```

Fig.5:Destroye All Window

RESULTS & DISCUSSION

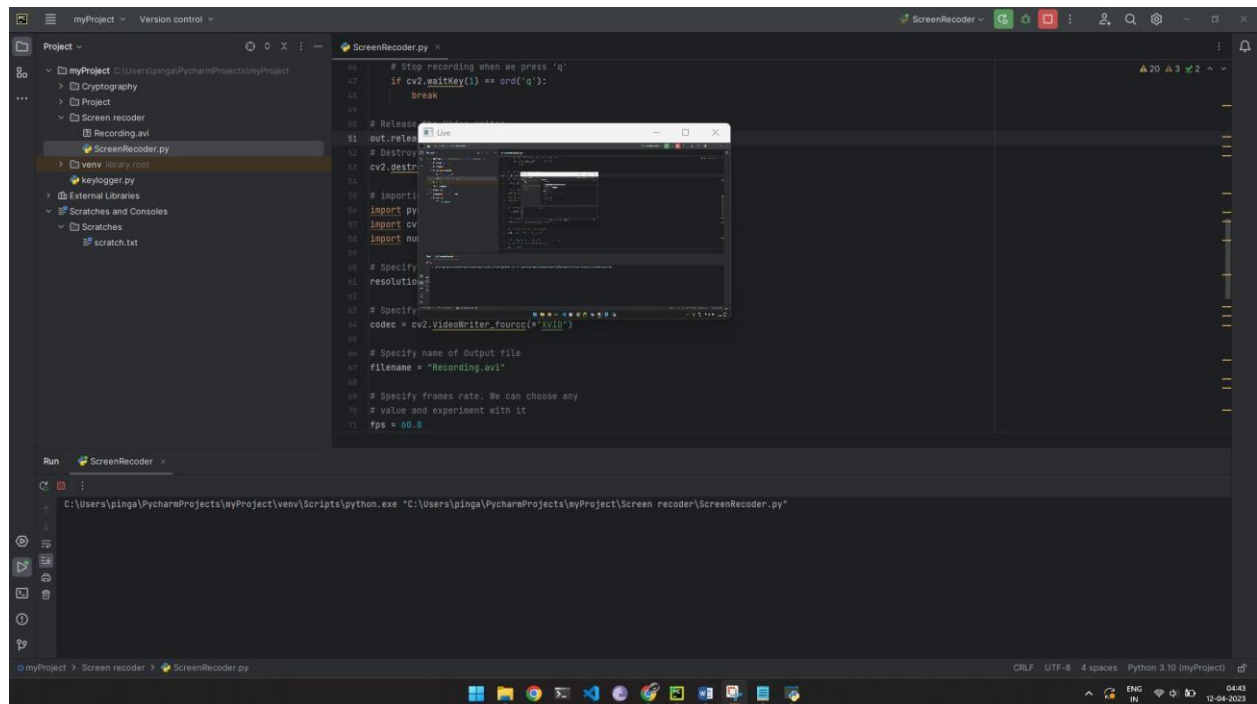


Fig.6:Result

CONCLUSION

Screen recording technology has come a long way from its inception in the early 2000s, and it continues to evolve and impact various industries in the present day. In the past, screen recording was a simple process that required basic software and low processing power, and it was primarily used for capturing desktop videos for personal or educational purposes.

Today, screen recording technology has advanced significantly, driven by the increasing need for remote work, online learning, and other digital activities. Modern screen recording software is designed to capture high-quality videos with customizable settings, editing options, and real-time collaboration features. Screen recording technology is now widely used in various industries such as marketing, e-learning, customer support, and software development to create engaging visual content, improve communication, and document work processes.

Looking to the future, screen recording technology is likely to continue advancing with the integration of augmented reality (AR) and virtual reality (VR), real-time collaboration, artificial intelligence (AI) and automation, improved video quality, and cloud-based storage and sharing. These advancements have the potential to greatly benefit industries by improving communication, collaboration, training, and documentation.

In conclusion, screen recording technology has come a long way since its inception, and it will continue to advance and impact various industries in the future. From simple desktop videos to high-quality and collaborative video content, screen recording technology has become an essential tool for modern digital Activities.

APPENDIX

Main programme where all the works are performed:

```
# importing the required packages
import pyautogui
import cv2
import numpy as np

# Specify resolution
resolution = (1920, 1080)

# Specify video codec
codec = cv2.VideoWriter_fourcc(*"XVID")

# Specify name of Output file
filename = "Recording.avi"

# Specify frames rate. We can choose
# any value and experiment with it
fps = 60.0

# Creating a VideoWriter object
out = cv2.VideoWriter(filename, codec, fps, resolution)

# Create an Empty window
cv2.namedWindow("Live", cv2.WINDOW_NORMAL)

# Resize this window
cv2.resizeWindow("Live", 480, 270)

while True:

    # Take screenshot using PyAutoGUI
    img = pyautogui.screenshot()

    # Convert the screenshot to a numpy array
    frame = np.array(img)

    # Convert it from BGR(Blue, Green, Red) to
```

```

# RGB(Red, Green, Blue)
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Write it to the output file
out.write(frame)

# Optional: Display the recording screen
cv2.imshow('Live', frame)

# Stop recording when we press 'q'
if cv2.waitKey(1) == ord('q'):
    break

# Release the Video writer
out.release()
# Destroy all windows
cv2.destroyAllWindows()

# importing the required packages
import pyautogui
import cv2
import numpy as np

# Specify resolution
resolution = (1920, 1080)

# Specify video codec
codec = cv2.VideoWriter_fourcc(*"XVID")

# Specify name of Output file
filename = "Recording.avi"

# Specify frames rate. We can choose any
# value and experiment with it
fps = 60.0

# Creating a VideoWriter object
out = cv2.VideoWriter(filename, codec, fps, resolution)

# Create an Empty window
cv2.namedWindow("Live", cv2.WINDOW_NORMAL)

```

```

# Resize this window
cv2.resizeWindow("Live", 480, 270)

while True:
    # Take screenshot using PyAutoGUI
    img = pyautogui.screenshot()

    # Convert the screenshot to a numpy array
    frame = np.array(img)

    # Convert it from BGR(Blue, Green, Red) to
    # RGB(Red, Green, Blue)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Write it to the output file
    out.write(frame)

    # Optional: Display the recording screen
    cv2.imshow('Live', frame)

    # Stop recording when we press 'q'
    if cv2.waitKey(1) == ord('q'):
        break

# Release the Video writer
out.release()

# Destroy all windows
cv2.destroyAllWindows()

```

BIBLIOGRAPHY

1. Screen Recording for User-Centered Design: A Guide for Designers and Developers, edited by Antti Knutas, Marko Nieminen, and Ilpo Koskinen.
2. "Screen Recording Technology for e-Learning: A Review," by J. Kim and J. Lee, published in the International Journal of Emerging Technologies in Learning (IJET).
3. "Screencasting: A New Tool for Online Education," by A. L. Rose, published in the Journal of Interactive Online Learning.
4. "A Review of Screen Recording Tools for Instructional Design," by J. G. Lee, published in the International Journal of Emerging Technologies in Education.
5. "Screen Recording in Software Development: A Review of Best Practices and Tools," by M. C. Canright and A. R. Shaw, published in the Journal of Software Engineering and Applications.