Plagiarism - Report

Originality Assessment

28%

Overall Similarity

Date: Apr 30, 2024
Matches: 1202 / 4226 words
Sources: 30

Remarks: Moderate similarity detected, you better improve the document (if needed).

Verify Report:

Revolutionizing Blind Navigation

Through AI Voices

Abstract— Navigating the complexities of aiding visually impaired individuals in object detection and environmental navigation is undeniably challenging. Leveraging the latest advancements in artificial intelligence, our project aims 10 to create a robust system capable of recognizing objects and delivering essential audio feedback to the user. The core of our implementation focuses on providing audio instructions, furnishing vital information about objects directly ahead, and intuitively guiding blind individuals in their chosen direction.

Index Terms—Accessibility technology, AI, Audion guidance system, Accessibility technology.

I. INTRODUCTION

Introducing our groundbreaking project, we are dedicated to revolutionizing the landscape of assistance 10 for visually impaired individuals. By harnessing cutting-edge advancements in artificial intelligence, our mission is to develop a sophisticated system adept at object recognition and environmental navigation. At the heart of our endeavor lies the commitment to delivering crucial audio feedback to users, empowering them with essential information about their surroundings. With a primary focus on providing intuitive guidance and detailed descriptions of objects directly ahead, our project aims to significantly enhance the autonomy and safety of blind individuals in their daily navigation endeavors..

II. System Analysis

A. Existing System

In these paper author explained us about object detection from the image and represented it by their name and speech. 2 And it also helps the blind people in location and encoded the audio into 2 channel audios with the help of 3D binaural sound. In these 1 a video is capture with portable camera device from client side and it is streamed to a server for real time image recognition with object detection. [2] which mean it identify and follow the same object in sequence of video frames sometimes video may have some noise. [8] to remove that noise from frames noise reduction technique is used that improve the image quality and extraction of object frame is used to detected the object based on color of the moving frame. using different feature extraction of object from frame is used they are called object detectionFinal Stage

B. Disadvantages

1. The cost of wearable devices with the necessary sensors and cameras may be high, making the system inaccessible to some blind individuals

2. The system's reliability may [1] be affected by various environmental factors, such as lighting conditions, weather, and obstacles.

3. The wearable devices may require frequent maintenance and updates, adding to the overall cost of the system.

4. Limited battery life and charging requirements: Wearable devices for blind individuals often rely on batteries to power the sensors and cameras.MATH

C. Proposed System

1. The proposed system uses wearable [6] devices with a camera, sensors, and audio output for object detection.

2. YOLO algorithm is used for real-time object detection and classification, such as people, vehicles, and obstacles.

3. The system relays [26] the detected object information to the blind person through an audio output and/or haptic feedback, providing navigation assistance to help them navigate their surroundings more safely and independently.

4. [2] This project tries to detect the object and transform that object into the audio form and inform blind person about those objects.

5. Our system consists of a box which has a portable camera and a system which will process that image. image are captured with a portable camera device with real-time image recognition on existing object detection models. after detecting an object that information is translate into audio.

D. Advantages of Proposed System

1) Increased independence and safety

2) [1] Real-time object detection and classification using YOLO algorithm

3) Customizable audio and/or haptic feedback

4) Navigation assistance feature

5) Machine learning capabilities 6. Integration with other devices.

III. System Architecture

User Interface: [6] This is the entry point for the system, where users interact with the application. It could be [20] a graphical user interface (GUI) or a command-line interface (CLI), depending on the design of the application.

Web Application: The user interface communicates with the web application, which is responsible for handling user requests and responses over the internet. It provides a platform-independent interface accessible through a web browser.

Flask Application: Flask is a lightweight [22] web framework for Python. In this architecture, the Flask application acts as the backend server for the web application. It handles HTTP requests from the web application, processes them, and generates appropriate responses.

Object Detection Module: [1] This module is responsible for detecting objects in images or video frames. It uses computer vision techniques, such as deep learning models like YOLO (You Only Look Once), to identify and localize objects within the input data.

AI Voice Assistance Module: Once objects are detected, this module provides audio feedback or instructions to the user based on the detected objects. It uses artificial intelligence (AI) techniques, such as natural language processing (NLP) and speech synthesis, to generate human-like voice responses.

In summary, the system architecture starts [16] with the user interface, which communicates with the web application. The web application interacts with the Flask backend, which houses the object detection and AI voice assistance modules. [1] The object detection module identifies objects in images or video, while the AI voice assistance module provides audio feedback to the user based on the detected objects. This architecture enables real-time [6] object detection and voice-based guidance for users interacting with the system through a web interface.Helpful Hints

IV. Python modules

A. OpenCV2

[13] OpenCV, short for Open Source Computer Vision Library, stands as a prominent open-source platform catering to a broad spectrum of functionalities concerning image and video processing, object detection, and machine learning. Initially conceived by Intel in 1999, it has evolved into a collaborative effort maintained by a diverse community of developers. OpenCV 2, marking a significant milestone, continues [3] to be a pivotal tool across numerous domains, including robotics, surveillance, augmented reality, and medical imaging. Equipped with an extensive repertoire of pre-built algorithms, it facilitates tasks ranging from image filtering and edge detection to face detection and object tracking. Moreover, OpenCV 2 extends its utility by supporting multiple programming languages, notably Python, C++, and Java, thereby offering versatility and accessibility to developers worldwide.

B. Numpy

NumPy, an abbreviation for Numerical Python, serves as a pivotal Python library indispensable for numerical computations. It introduces a robust array object capable of executing intricate mathematical operations effortlessly. [1] In contrast to conventional Python lists, NumPy arrays exhibit superior speed and efficiency, rendering them invaluable for scientific computing, data analysis, and machine learning endeavors. Moreover, NumPy boasts an extensive suite [11] of mathematical functions and utilities tailored for tasks spanning linear algebra, Fourier analysis, and statistical analysis. As a cornerstone library for scientific computing in Python, NumPy enjoys widespread adoption across diverse domains including physics, chemistry, engineering, and finance, solidifying its status as an essential tool for researchers and practitioners alike.

C. PYTTSX3

Pyttsx3, a Python library renowned for its text-to-speech conversion capabilities, furnishes a user-friendly interface simplifying the conversion of [1] text into speech. Catering to a multitude of languages, Pyttsx3 facilitates the generation of speech output for diverse applications including virtual assistants, automated customer service systems, and accessibility features. Leveraging the robust foundation [29] of the Microsoft Speech API and the SAPI5 interface, Pyttsx3 ensures high-quality speech synthesis. Additionally, Pyttsx3 offers comprehensive support for various parameters such as voice type, rate, volume, and pitch, empowering users to customize speech outputs according to their preferences and requirements.

D. Matplotlib

Matplotlib, a Python library renowned for data visualization, boasts an array of tools catering to the creation of 2D and 3D plots, histograms, scatterplots, and beyond. Offering extensive [14] customization options, Matplotlib empowers users to craft professional-grade graphics suitable for scientific publications, business reports, and presentations. Supporting diverse output formats like [19] PNG, PDF, SVG, and EPS, Matplotlib seamlessly integrates with various GUI frameworks including Tkinter, PyQt, and wxPython. Widely embraced across domains encompassing data science, engineering, finance, and biology, Matplotlib stands as an indispensable tool for visualizing data and conveying insights effectively.

E. GTTS :

GTTS, short for Google Text-to-Speech, is a Python library harnessing Google's Text-to-Speech API to seamlessly transform text into speech. With its user-friendly interface and extensive language support, GTTS caters to a myriad of applications including automated voice notifications, audiobooks, and podcasts. Built atop the requests and gTTS-token libraries, GTTS effortlessly manages HTTP requests and token generation essential for accessing the Google Text-to-Speech API, ensuring smooth operation and reliable performance.

F. Playsound

Playsound is a [11] Python library designed for playing audio files, offering a straightforward and intuitive interface for playing both WAV and MP3 files. Its cross-platform compatibility ensures seamless operation [27] across Windows, macOS, and Linux environments. Leveraging the simpleaudio and pydub libraries, Playsound efficiently manages audio playback and file format conversion tasks, facilitating smooth and hassle-free audio playback experiences.

V. Module Description

1. Image Acquisition

2. Preprocessing

3. Feature Extraction

4. Object Detection

5. Text to Voice

A. Image Acquisition

Image acquisition is the initial step in [1] the process of obtaining images for subsequent analysis and processing. It involves capturing unprocessed images from external sources, such as cameras or sensors, and converting them into a digital [10] format that can be manipulated and analyzed by computer algorithms.

The process typically begins [1] with the use of optical devices, such as cameras or sensors, to capture images of objects or scenes. These optical devices capture light reflected or emitted by the objects and convert it into electrical signals. These signals are then digitized using analog-to-digital converters (ADCs), resulting in a digital representation of the captured image.

Once [1] the image is in digital format, it can be stored, processed, and analyzed using various software tools and algorithms. Image acquisition [23] plays a crucial role in fields such as medical imaging, remote sensing, surveillance, and computer vision, where high-quality images are essential for accurate analysis and decision-making.

B. Preprocessing

Pre-processing is a crucial stage in image processing where [1] the input image undergoes various cleaning and enhancement tasks to improve its quality and suitability for further analysis. The primary objective of pre-processing is to enhance the image by removing noise, correcting distortions, and standardizing its format for subsequent processing steps.

1. Noise Removal: The first step in pre-processing is often noise removal. Noise can distort [1] the image and affect the accuracy of subsequent processing algorithms. Common techniques for noise removal include median filtering, Gaussian filtering, and morphological operations such as erosion

and dilation.

2. Image Enhancement: After noise removal, image enhancement techniques may be applied to improve the visual quality of the image. This can include adjusting brightness and contrast, sharpening edges, and enhancing details using techniques such as histogram equalization.

3. Color Space Conversion: Depending on [1] the requirements of the application, the image may need to be converted to a different color space. For example, converting a color image to grayscale simplifies subsequent processing tasks and reduces computational complexity. Other color space conversions may also be performed based on the specific requirements of the application.

4. Image Binarization: In many cases, it is necessary to convert the image into a binary format, where each pixel is classified as either foreground or background. This is often achieved through image binarization techniques, such as thresholding, where a threshold value is applied to the pixel intensities to separate the foreground objects from the background. Binarization simplifies subsequent processing [3] tasks, such as object detection and segmentation, by reducing the complexity of the image.

Overall, pre-processing plays a vital role in image analysis [1] and computer vision tasks by improving the quality of input images, reducing computational complexity, and enhancing the effectiveness of subsequent processing algorithms.

C. Feature Extraction

[24] Feature extraction is a critical step in data preprocessing and analysis, particularly in fields such as image processing, pattern recognition, and machine learning. It involves transforming raw data, often represented by a large number of variables or features, into a reduced set of meaningful and representative features that capture the essential information contained in the original data.

1. Dimensionality Reduction: [1] One of the primary objectives of feature extraction is to reduce the dimensionality of the data. High-dimensional data, characterized by a large number of variables, can be computationally expensive to process and may suffer from the curse of dimensionality. Feature extraction [5] techniques aim to overcome this challenge by selecting or transforming variables into a lower-dimensional space while preserving as much relevant information as possible.

2. Selection and Combination of Variables: Feature extraction methods typically involve [12] selecting a subset of variables from the original data or combining them to create new features. This selection process is guided by various criteria, including relevance [5] to the task at hand, discriminative power, and computational efficiency. Common techniques [12] for feature selection include filter methods, wrapper methods, and embedded methods, each with its own advantages and limitations.

3. Preservation of Information: Despite [5] reducing the dimensionality of the data, feature extraction techniques strive to retain as much relevant information as possible. This involves identifying features that capture the inherent structure, patterns, and variability present in the original data. By focusing on informative features and discarding redundant or irrelevant ones, feature extraction helps improve the efficiency and effectiveness of subsequent data analysis tasks.

4. Representation of Original Data: The ultimate goal of [1] feature extraction is to create a compact and informative representation of the original data. This representation, [14] often referred to as feature space, encapsulates the essential characteristics of the data while minimizing redundancy and noise. It facilitates the interpretation, visualization, and analysis of complex data sets and enables [10] the development of efficient and accurate machine learning models.

[5] In summary, feature extraction plays a crucial role in transforming raw data into a more manageable and informative form, enabling more efficient and effective data analysis, pattern recognition, and decision-making processes. By selecting, combining, and transforming variables, feature extraction techniques help uncover underlying patterns and structures in the data, leading to improved insights and predictive performance.

D. [1] Object Detection

The YOLO (You Only Look Once) algorithm is a groundbreaking approach to object detection in images. Unlike traditional methods that rely on multiple stages and complex processing pipelines, YOLO operates as a single neural network that predicts bounding boxes and class probabilities directly from the full image in a single evaluation. Here's an expanded explanation of how YOLO works:

1. Feature Extraction and Matching: The first step in the YOLO algorithm involves extracting features from the input image using a convolutional neural network (CNN). These features are then used to

match with the training objects or classes that the model has been trained on. The CNN learns to capture meaningful patterns and representations from the image data, which are essential for accurate object detection.

2. Single-Stage Object Detection: YOLO belongs to the category of single-stage object detection models, which means that it directly predicts bounding boxes and class probabilities without the need for separate region proposal and classification stages. This streamlined approach allows YOLO to achieve real-time performance, making it highly suitable for applications requiring fast and efficient object detection.

3. Unified Network Architecture: One of the key features of YOLO is its unified network architecture, where a single convolutional neural network is responsible for both object localization (predicting bounding boxes) and classification (predicting object classes). This architecture simplifies the model design and training process, leading to faster inference times and improved performance.

4. Efficiency and Speed: YOLO is renowned for its speed and efficiency compared to traditional object detection methods like Faster R-CNN. By processing the entire image at once and leveraging shared convolutional layers, YOLO can generate predictions swiftly, making it well-suited for real-time applications such as video surveillance, autonomous driving, and robotics.

5. Grid-Based Prediction: YOLO divides the input image into a grid of cells and predicts bounding boxes and class probabilities for each cell. Each cell is responsible for detecting objects that fall within its spatial region, resulting in a grid-based prediction mechanism. Despite generating a large number of bounding boxes initially, YOLO employs techniques such as non-maximum suppression to consolidate overlapping predictions and produce the final set of detected objects.

In summary, the YOLO algorithm revolutionizes object detection with its single-stage architecture, unified network design, and real-time performance. By directly predicting bounding boxes and class probabilities from the entire image, YOLO offers unparalleled speed and efficiency, making it a popular choice for various computer vision applications..

E. Voice Output

The process of converting text to speech (TTS) using the GTTS (Google Text-to-Speech) library in Python involves several steps. Here's an expanded explanation of how GTTS works to produce voice output in headphones:

1. Text Input: The first step is to provide the text that you want to convert to speech. This text can be in any language supported by GTTS, and it can include punctuation, special characters, and even emojis.

2. GTTS Library: Once you have the text input, you use the GTTS library in Python to convert this text into speech. GTTS is a Python library that interfaces with Google's Text-to-Speech API, allowing you to generate speech output from text strings.

3. Text-to-Speech Conversion: When you pass the text input to the GTTS library, it sends a request to Google's Text-to-Speech API, which processes the text and generates an audio file containing the speech output. Google's API uses advanced speech synthesis techniques to produce high-quality and natural-sounding speech.

4. Audio Output: Once the audio file is generated by the Google Text-to-Speech API, it is downloaded to your local machine. You can then play this audio file using any audio playback mechanism supported by Python. In the case of headphones, you would typically use a library like `playsound` or `pygame.mixer` to play the audio file and route the output to the headphones connected to your computer.

5. Headphone Output: By default, the audio output generated by the GTTS library is played through the default audio output device on your computer. However, you can specify the audio output device to be your headphones if they are connected to your computer. This ensures that the speech output is heard only through the headphones and not through the computer's speakers.

Overall, the GTTS library simplifies the process of converting text to speech in Python and provides a convenient way to generate speech output for various applications, including virtual assistants, accessibility features, and audio notifications. By leveraging Google's Text-to-Speech API, GTTS delivers high-quality speech synthesis with support for multiple languages and customizable parameters.

VI. Algorithms

A. YOLO(You Only Look Once) Algorithm

The YOLO (You Only Look Once) algorithm, introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, revolutionized real-time object detection with its innovative approach. Here's a detailed explanation of how the YOLO algorithm works and its key advantages:

1. Single [1] Convolutional Neural Network (CNN): YOLO is based on a single CNN trained end-to-end to predict bounding boxes and class probabilities of objects in an input image.

Unlike [4] traditional object detection methods that rely on complex pipelines or region proposal networks, YOLO processes the entire image in a single pass.

2. Grid-based Prediction: [7] YOLO divides the input image into a grid of cells and each cell is responsible for predicting the objects contained within its region. [25] This grid-based approach allows YOLO to efficiently localize objects without the need for extensive post-processing.

3. [3] Bounding Box Prediction: For each grid cell, YOLO predicts multiple bounding boxes along with the corresponding confidence scores and class probabilities. These bounding boxes represent the [1] regions in the image where objects are detected, and the confidence scores indicate the likelihood of the presence of an object.

4. [7] Non-Maximum Suppression (NMS): After generating bounding boxes for all grid cells, YOLO applies NMS to eliminate duplicate detections and refine the final set of predictions. NMS ensures [4] that only the most confident bounding boxes are retained, leading to accurate and reliable object detection results.

5. Speed and Accuracy: One of the key advantages of YOLO is its speed and efficiency. By using a single CNN for object detection, YOLO achieves real-time performance, making it suitable for applications requiring rapid processing, such as autonomous driving and surveillance systems.

Additionally, YOLO maintains high accuracy levels, rivaling or surpassing [1] traditional object detection methods while significantly reducing computational overhead.

6. Versatility and Applications: [28] YOLO has found widespread adoption across various domains, including autonomous vehicles, video surveillance, augmented reality, and medical imaging. Its versatility, [4] speed, and accuracy make it a valuable tool for tasks requiring robust object detection capabilities in real-world scenarios.

In summary, the YOLO algorithm represents a groundbreaking approach to object detection, offering a unique combination of speed, accuracy, and efficiency that has propelled it to the forefront of the field. [7] Its ability to process images in real-time with minimal computational resources makes it an indispensable tool for a wide range of applications in computer vision and beyond.

B. RCNN (Region – Based [1] Convolutional Neural Network) Algorithm

The RCNN (Region-based Convolutional Neural Network) algorithm, pioneered by Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik in 2014, represents a significant advancement in object detection. Here's an in-depth exploration of the RCNN algorithm and its key attributes:

1. Multi-Stage Process: The RCNN algorithm follows a multi-stage approach comprising region proposals, feature extraction, and object classification. Initially, it generates region proposals using selective search or a similar method to identify potential object locations in the input image.

2. Feature Extraction: Once the [18] region proposals are obtained, the RCNN algorithm extracts features from each proposed region using a pre-trained convolutional neural network (CNN), typically based on architectures like AlexNet or VGGNet. These features are then [1] used to represent the content of the proposed regions.

3. Object Classification: After feature extraction, the RCNN algorithm performs object classification using a separate classifier, such as a support vector machine (SVM), to determine the presence and category of objects within each region. This stage involves training the classifier on labeled data to learn discriminative features for object recognition.

4. Advantages of RCNN: One of the primary advantages of the RCNN algorithm is its high accuracy. By leveraging region proposals and fine-grained feature extraction, RCNN achieves impressive performance on [7] object detection tasks, surpassing earlier methods based on handcrafted features. Additionally, RCNN offers flexibility, [14] as it can be enhanced with various improvements and extensions.

5. Variant Enhancements: Over time, the RCNN algorithm has been refined with several variants to address its limitations and improve its performance. These variants include Fast R-CNN, which accelerates the algorithm by sharing convolutional features across region proposals, and Faster R-CNN, which introduces [30] a region proposal network (RPN) to generate region proposals more efficiently.

6. Applications and Impact: The RCNN algorithm and its variants have been extensively applied in diverse fields 2 such as autonomous vehicles, surveillance systems, robotics, and medical imaging. 7 Their ability to accurately detect objects in images has led to advancements in areas requiring robust computer vision capabilities.

In summary, the RCNN algorithm 4 stands as a cornerstone in the evolution of object detection, offering a robust framework for accurately identifying objects in images through a multi-stage process. Its high accuracy and adaptability have cemented its status as a fundamental tool in the computer vision toolkit, driving innovation and progress across various domains.

VII. Output Screenchots

A. Flask Code Running

B. Home Page

C. About Page

D. Contact Page

E. HelpPage

F. Execution Page

G. Person Output

H. Water Bottle Output

VIII. Conclusion

[1] Object detection with audio feedback is a useful application of computer vision and speech recognition technologies. By detecting objects in real-time and providing an audio description of the [2] object's class and confidence level, it can assist people who are visually impaired in navigating their surroundings and identifying objects.

In the code provided, the YOLO [1] CNN was used to detect objects from a video stream, and the detected object's class label and confidence level were passed to the pyttsx3 text-to-speech engine to provide audio feedback. The code [6] can be modified to detect a specific object, such as a cellphone, by adjusting the CLASSES list and confidence threshold.

Overall, [3] object detection with audio feedback has the potential to improve the accessibility and independence of people with visual impairments, and it's a great example of how computer vision and speech technologies can work together to create innovative solutions.

IX. Future Work

Looking ahead, [1] there are several avenues for further enhancement and expansion of our revolutionary system. Firstly, we aim to incorporate [2] machine learning techniques to continuously improve object detection accuracy and optimize voice assistance functionality. Additionally, we plan [6] to explore the integration of advanced sensors and wearable devices to enhance real-time environmental awareness for users. Moreover, extending the geographical coverage and language support of our system will broaden its accessibility and impact globally. Furthermore, collaborating with experts in [5] the field of accessibility technology and conducting user-centered design research will ensure the ongoing relevance and effectiveness of our solution. Ultimately, our commitment to innovation and inclusivity drives our pursuit of continually pushing the boundaries [10] of assistive technology to empower and enrich the lives of visually impaired individuals.

REFERENCES

[1] 1 Girshick, R., Donahue, J., Darrell, T., Malik, J., (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.

[2] Glorot, X., 17 Bengio, Y., (2010). Understanding the difficulty of training deep feedforward neural networks., in: Aistats, pp. 249–256.

[3] Goodfellow, I.J., (2013). Piecewise linear multilayer perceptrons and dropout. stat 1050, 22.

[4] He, K., Zhang, X., Ren, S., Sun, J., (2016). Deep residual 15 learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

[5] Jung, K., (2012). Object 21 recognition on mobile devices, in: Consumer Electronics-Berlin (ICCE-Berlin), 2012 IEEE International Conference on, IEEE. pp. 258–262.

1

Sources

1	https://link.springer.com/article/10.1007/s11042-022-13153-y INTERNET 9%
2	https://www.academia.edu/63399837/Object_Detection_and_Translation_for_Blind_People_Using_Deep_Learning INTERNET 3%
3	https://www.researchgate.net/publication/351411017_Real-Time_Object_Detection_Using_YOLO_A_Review INTERNET 3%
4	https://www.basic.ai/post/yolo-algorithm-for-object-detection#:~:text=As a breakthrough in the field of object,and locations of different objects in an image. INTERNET 3%
5	https://robots.net/fintech/what-is-feature-vector-in-machine-learning/ INTERNET 2%
6	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9324285/ INTERNET 1%
7	https://visionplatform.ai/yolov9/ INTERNET 1%
8	https://www.irjet.net/archives/V7/i3/IRJET-V7I3563.pdf INTERNET 1%
9	https://medium.com/@pelinokutan/how-to-convert-text-to-speech-with-python-using-the-gtts-library-dbe3d56730f1 INTERNET 1%
10	https://link.springer.com/article/10.1007/s41324-023-00522-4 INTERNET 1%
11	https://flexiple.com/python/numpy-tutorial INTERNET 1%
12	https://www.blog.trainindata.com/feature-selection-for-machine-learning/ INTERNET <1%
13	https://www.geeksforgeeks.org/opencv-python-tutorial/ INTERNET <1%
14	https://productdraft.com/enhancing-data-visualization-with-python-matplotlib-vs-seaborn/ INTERNET <1%

15 https://link.springer.com/article/10.1007/s12524-022-01506-x
INTERNET
<1%

16 https://integrio.net/blog/modern-web-application-architecture
INTERNET
<1%

17 https://proceedings.mlr.press/v9/glorot10a.html
INTERNET
<1%

18 https://medium.com/@kattarajesh2001/object-detection-part-2-two-stage-detectors-r-cnn-fast-r-cnn-faster-r-cnn-026e5fac5dd0
INTERNET
<1%

19 https://dev.to/emilossola/exploring-data-visualization-with-pythons-matplotlib-library-5amh
INTERNET
<1%

20 https://phoenixnap.com/kb/cli-vs-gui
INTERNET
<1%

21 https://www.itm-conferences.org/articles/itmconf/ref/2021/05/itmconf_icacc2021_03001/itmconf_icacc2021_03001.html
INTERNET
<1%

22 https://www.browserstack.com/guide/web-application-development-guide
INTERNET
<1%

23 https://ied.eu/blog/education-blog/empowering-minds-transformative-power-education/
INTERNET
<1%

24 https://scikit-learn.org/stable/modules/feature_extraction.html
INTERNET
<1%

25 https://medium.com/@sachinsoni600517/concept-of-yolov1-the-evolution-of-real-time-object-detection-d773770ef773
INTERNET
<1%

26 https://link.springer.com/chapter/10.1007/978-981-99-1624-5_57
INTERNET
<1%

27 https://medium.com/@seomaven979/building-cross-platform-desktop-apps-with-net-b6f8d277a6c9
INTERNET
<1%

28 https://www.augmentedstartups.com/blog/yolo-object-detection-a-comprehensive-guide-to-real-time-visual-recognition
INTERNET
<1%

29 https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ms723627(v=vs.85)
INTERNET
<1%

30	https://paperswithcode.com/method/faster-r-cnn INTERNET	<1%

EXCLUDE CUSTOM MATCHES OFF

EXCLUDE QUOTES ON

EXCLUDE BIBLIOGRAPHY ON