IMAGE RECOGNITION WITH CLOUD VISUAL RECOGNITIONS

Problem Definition :

The project involves creating an image recognition system using IBM Cloud VisualRecognition. The goal is to develop a platform where users can upload images, and theSystem accurately classifies and describes the image contents. This will enable users to craftEngaging visual stories with the help of AI-generated captions, enhancing their connectionWith the audience through captivating visuals and compelling narratives

Design Thinking:

1. Image Recognition Setup: Set up the IBM Cloud Visual Recognition service andObtain the necessary API keys.

2.User Interface: Design a user-friendly interface for users to upload images and viewThe AI-generated captions.

3.Image Classification: Implement the image classification process using the IBMCloud Visual Recognition API.

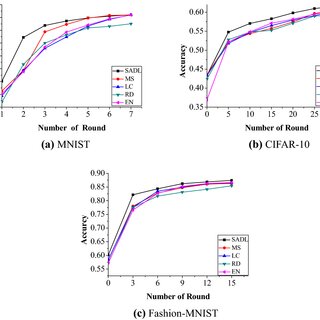
4.AI-Generated Captions: Integrate natural language generation to create captions forThe recognized images.

5.User Engagement: Design features to allow users to explore, save, and share theirAIenhanced images.

INTRODUCTION:

Code reviews are policy in many software development organizations, and it is commonlyBelieved that code reviews are an economical way to discover faults before a softwareProduct is deployed. Indeed, it is even suggested that code that has not been adequatelyReviewed has twice the faults of reviewed code. However, many software engineers areOverwhelmed with work, so proper code reviews are often not done. The reviewability ofSoftware is affected by many factors such as documentation, logic, semantics, and syntax.Source code includes aspects that might even be considered aesthetic, and aestheticAspects might turn tedious and possibly overwhelm the review process . In a paper byYazdani and Manovich, non-photographic images, such as screenshots and images of textMessages, were analysed and found they could be useful in predicting social trends . ThisPaper aims to evaluate the possibility of using “screenshots” of source code with machineLearning image recognition as part of the software code review process. Tools to reduceMonotonous tasks related to reviews could be very valuable. This paper begins byDiscussing the readability aspects of code and estimates the impact style has on reviews.We then created images of poorly styled code and properly styled code and used machineLearning to train an image recognizer to identify poorly formatted code and present positiveResults. Creating source code“screenshot images” for analysis could be part of automatingCode reviews. Using automation as part of the review process could make softwareEngineers more efficient.The diversity and high generationSpeed of license plate training sample set can achieve the purpose of effectively trainingStrong classifier. By using genetic algorithm to optimize BP neural network to classify licensePlate information, the anti-interference ability and license plate recognition accuracy areImproved to a certain extent.

**Result** **and** **data** **analysis** :



**Uses** **of** **Image** **Recognition** :

Drones:

Drones equipped with image recognition capabilities Can provide vision-based automatic monitoring, Inspection, and control of the assets located in remote Areas.

Manufacturing:

Inspecting production lines, evaluating critical Points on a regular basis within the premises. Monitoring the quality of the final products to Reduce the defects

AutonomousVehicles:

Autonomous vehicles with image recognition can Identify activities on the road and take necessary Actions. Mini robots can help logistics industries To locate and transfer the objects from one place To another. It also maintains the database of the Product movement history to prevent the product From being misplaced or stolen.

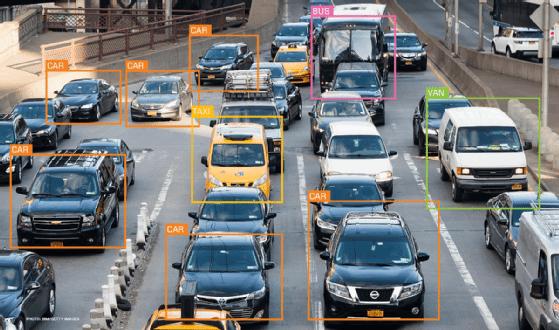
MilitarySurveillance:

Detection of unusual activities in the border areas and automatic decision-making Capabilities can help prevent infiltration and

Result in saving the lives of soldiers.

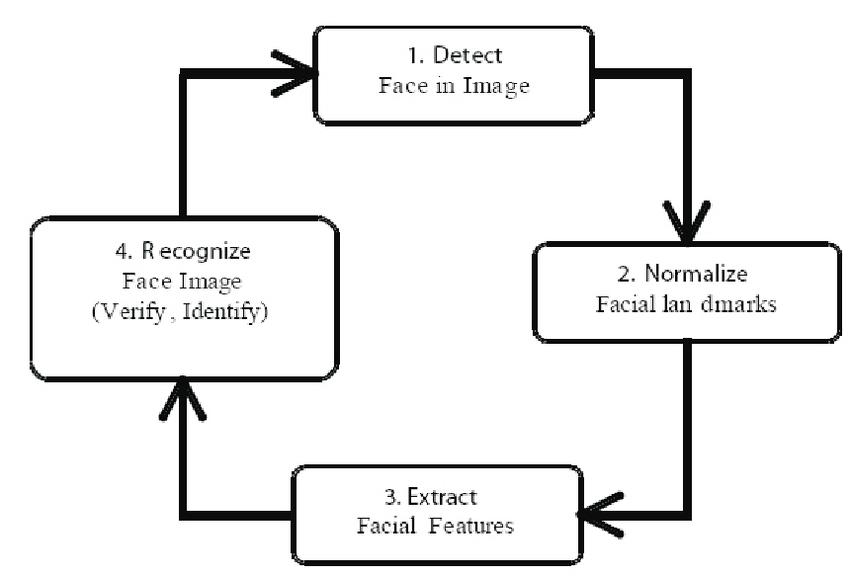
Forest Activities:

Unmanned Aerial Vehicles can monitor the Forest, predict changes that can result in forest Fires, and prevent poaching. It can also provide a Complete monitoring of the vast lands, which Humans cannot access easily.

**Image** **recognition** :



This paper begins byDiscussing the readability aspects of code and estimates the impact style has on reviewsWe then created images of poorly styled code and properly styled code and used machineLearning to train an image recognizer to identify poorly formatted code and present positiveResults. Creating source code“screenshot images” for analysis could be part of automatingCode reviews. Using automation as part of the review process could make softwareEngineers more efficient.

****

**Github ling:**

[**https://github.com/users/JabaCHristena/emails/280617661/confirm\_verification/44354**](https://github.com/users/JabaCHristena/emails/280617661/confirm_verification/44354)

**293?via\_launch\_code\_email=true**

**Coding:**

# This is a demo of running face recognition on live video from your webcam. import face\_rec0U1itiM1 import cv2 import nt—y as rp

# Get a reference to webcam (the default one)

# Load a sample picture and learn how to recognize it.

# Load a second sample picture and learn how to recognize it.

# Create arrays of known face encodings and their names

1 kncm. face n.s

Barack %a.a" ,

"Joe Biden"

# # Initialise some variables

face\_encodings = [J face\_rwnes = [J process\_this fr— = True

\*ile True:

# Grab a single frame of video ret, fr— = video\_capture.read() # Resize frame of video to 1/4 size snall\_fr— = cv2.resize(fr—, (O, O), fx=ø. 25, free. 25)

# Convert the image from BGR color to RGB color (which face\_recognition uses)

if process\_thi s\_fr— :

# Find all the faces and face encodings in the current frame of video face locatims

def fer\_simple\_cnn()

# model = A)dels. Sequential( ) model.add(Conv2D(32, (3, 3), activation= relu• , 48,

model.add (MaxP001ing2D( (2, 2)) ) model. (3, 3), activation= relu' ) ) model. add (MaxPooIing2D( (2, 2)) ) model.add (Conv2D(64, (3, 3), activation= relu ) ) model. add (Flatten( ) ) model.add (Dense (64, activaticm= • relu• ) ) model.add (Dense (7, activation= • softmax ))

return

Thank you.