code.sprint MT

TASK BOOKLET

Software Developers Category

2022

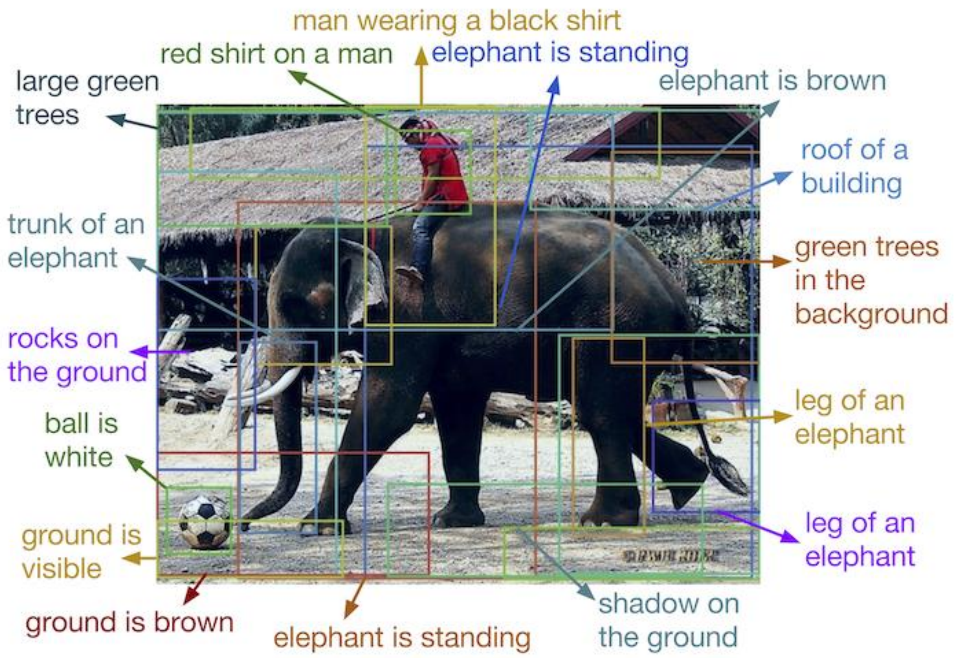
ICE Malta

GOVERNMENT OF MALTA
RESEARCH AND INNOVATION
MINISTRY FOR EDUCATION, SPORT, YOUTH
DIRECTORATE FOR LEARNING AND ASSESSMENT PROGRAMMES

EyeSmart

Hello!

Welcome to the CODESPRINT 2022 competition. During these two days, you will be flexing your coding and UI/UX design muscles to create a visual image processing application.



1. Design Brief

The Ministry for Education and Employment wishes to develop an application to use during expos and roadshows to show students in Malta the power of AI, and hopefully get them excited to start learning AI themselves. This year, the Ministry has decided to focus on computer vision.

Whilst computer vision algorithms are sophisticated and can take years to develop, many providers have started democratizing this technology and making it available to the general public. Examples include the Amazon Rekognition service, Microsoft Computer Vision API, Google Cloud Vision API, OpenCV and more. Using these APIs, developers can quickly build applications using advanced technologies, to 'wow' students and get them interested in AI.

The Ministry has developed a table of the core functionality required in the app, which is shown below.

Functionality		Notes
1	**Ability to recognize a face**	A face should be learnt by the system. On subsequently scanning the face, the system should greet the user by the name associated with the previously learnt face.
2	**Analyse a face**	The app should be able to recognize several attributes of a face, such as: • The face gender. • Approximate age. • Whether or not the face is smiling. • Whether or not eyes are open. • Whether or not the face has a beard or moustache. • Whether or not the mouth is open. • Whether or not glasses are being worn. • A guess as to the emotion being shown.
3	**Recognise Text**	The system should be able to recognize text (printed, handwritten, on clothing, on items etc.) and display it.
4	**Recognise Items**	The system should be able to recognize items in the image, such as humans, shelving, books, mobile phones etc.
5	**Display Technical Data**	Ideally, the system should display a map of the points of a face/image which have been scanned by the AI.
6	**Talk**	Ideally, responses from the system should be spoken by the system.

2. Technical Guidelines

The method you choose to implement the app is up to you. However, the following technical guidelines are intended to help ensure you stay on the right track.

2.1 Reference Implementation

The judging panel has created a reference implementation of this app in 16 hours. This is to ensure that the task given is possible within the timeframe allocated. A video of this app in operation is available below. We highly recommend that you watch this video carefully, to get an idea of the functionality and level of polish the judging panel is expecting.

https://codesprint2022.s3.eu-central-1.amazonaws.com/recognice.mp4

2.2 Platform

Your app can run on any platform you choose (Windows, macOS, Linux, iOS, Android, Web). We suggest creating a web application to make demonstration of the app easier during events, however we'll ultimately leave this up to you.

2.3 Development Environment

You are free to use ANY programming language you wish to create your solution. However, to remember that the solution must run on the judge's computers, and that you must provide both a binary/executable solution, as well as source code.

You are strongly advised to use existing computer vision APIs. Links are provided below:

- AWS Rekognition - https://aws.amazon.com/rekognition/
- Microsoft Computer Vision API - https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/#overview
- OpenCV – https://opencv.org
- Google Cloud Vision API - https://cloud.google.com/vision/
- Azure Face API - https://azure.microsoft.com/en-us/services/cognitive-services/face/#overview
- Clarifai - https://www.clarifai.com/computer-vision

If you're unsure what to use, try AWS Rekognition, which has a free plan and a relatively easy to use API with all capabilities required for this task.

2.5 Name

EyeSmart is a sample name – you are free to call your app whatever you want 😀

3. Judgement Criteria

Your submission will be given a maximum of 185 points. The criteria by which points are awarded are detailed below. **Note that you do not need to achieve all the criteria**, however, the more criteria you achieve, the greater your chances of winning! The numbers in **[brackets]** refer to the functionality in the design brief.

Criterion	Notes	Max Points
Core Functionality		
[1] Ability to recognize a face once it has been learnt.	Will be awarded based on how well faces can be recognized, as well as recognition speed.	**20**
[2] Ability to analyze a face and identify at least 5 attributes	Attributes can include gender, age, smile, eyes open, facial hair, mouth open, glasses, emotion. However, other attributes will also be considered.	**20**
[3] Ability to recognize text	System should be able to recognize text in various situations, on various items, and include handwritten or decorated text.	**20**
[4] Ability to recognize items	System should recognize various items in an image (unless the image simply contains one item with a blank background).	**20**
[5] Display technical data	System should show a map of plotted points/areas within a scanned image.	**10**
[6] Ability to talk	System should synthesize voice replies	**10**
UI/UX		
Neat/Aesthetically pleasant user interface	Rather than 'flair', we are looking for a neat, organized and functional UI	**10**
App is easy to use	The user should not need a manual to use the app	**5**
Experience	The User Experience is up to you, however, make sure your app: • Displays a live feed from the camera. • Displays a log of previous results. • Easily allows users to teach the system a face.	**15**
Code Quality		
Code is organized into packages/modules/units etc.		**5**
Separation between presentation and logic layers	For example, using a REST API model	**10**
Consistent and correct use of a programming paradigm	Such as OOP, AOP, functional etc.	**5**
Function cohesion	Functions should be kept small, and do one thing, without being too dependent on other functions	**5**
Inline documentation	i.e. comments	**5**
Maintainable code	Ex: use of abstract classes, interfaces, function prototypes etc. Depending on the programming paradigm chosen	**5**

Additional Functionality/Features		
Sample data is persistently stored	Faces learned by the system should be remembered on next using the system	**5**
Additional features	Additional features over and above the design brief will be graded, up to a maximum of 15 points	**15**

Submission Criteria

At the end of the time allocated to this competition, you must submit your code to the judging panel. The code, including all assets and other resources, must be submitted as a folder or compressed archive.

You will also be required to demonstrate your application running.

NOTES

eSkills
eSkills Malta Foundation

LeoVegas

FreeHour