**Tasks to Do**

1.Currently the website supports manually uploading the image from your computer so we will have to add a functionality which will allow users to directly upload pictures from their cameras.

2.Using Css we will have to style our website so that it looks aesthetically pleasing.

3. Currently the program does not check whether the file uploaded is of a valid extension like jpg,png . When submitting a file other than an image it throws an error, it would be better if the site could display the message like a received a file of invalid type expected image got pdf.

4.. Update this doc if you complete a task or feel like there are things still left to be done.

**Few Points about the Project**

The aim of the project was to create an app or website to classify the image of parts to 5 different B vehicles and further classify whether the part is defective or not.

# **Approach for the project**

The approach was to create a program which will work in 3 parts

1. Part 1 was to classify what is this image as in which part it is like is it a spark plug ,tire or engine . Since these are parts which can be easily distinguished or at least that is my assumption so we can use a model which is trained on lower resolution images and it will be able to give accurate results even if data is less.
2. Part 2 was to classify which vehicle this part belong to for this as was mentioned in ppt that there could be some parts which are not that distinguishable hence they require finer image classification for example engine for maruti will not very much differ from that of bolero (Assumption) hence the model is trained on higher resolution images.
3. Part 3, was to classify whether the part is defective or not.Again following from the same logic as point 2 we have used a model which is trained on higher resolution images.

# **Code for the project**

The main part of the program is the predict function which takes the image path as an argument and returns the prediction to display on the site along with the image uploaded.The code for preprocessing of the image is written keeping the car defect model in mind. More code is to be written which will first send the image to model of part 1 which will predict which part is it then the image is passed on to part2 and part3 model which are trained specifically on that part’s images and this returns which vehicle this part belongs to and whether it is defective or not.

# **Overall flow of Program**



After part 1 lets say the model identifies the image as spark plug then it calls the part 2 and part 3 model which is trained only on the images of spark plug(both defective and not defective) to classify it into 5 B vehicles. Note that part 1 model is trained on all the 200 parts but part 2 and 3 models are trained on specific parts hence they will be different for different parts.After part 2 identifies which B vehicle it belongs to, simultaneously the part 3 model which is trained on defective and non defective spark plug identifies whether the part is defective or not and together they display the result which is shown on the site.

**Models used for this project and their use**

1. **Intel Model:**This model is trained on Intel image classification data-set which contains images of scenery around the world like glaciers,buildings,streets etc. (Böttinger).The image shape used is (150,150) .This model can be used for part 1 of the project .This model is trained on a pretrained model of VGG19 with weights taken from imagenet.
2. **Oxford Model:**This model is trained on Intel image classification data-set which contains images of 17 flowers like daffodils,lilies etc. (Oxford).The image shape used is (224,224) .This model can be used for part 2 of the project .This model is trained on a pretrained model of VGG19 with weights taken from imagenet.
3. **Car defect detector model:**This model is trained on a data-set which contains images of cars with and without defects(Couldn't find reference ).The image shape used is (224,224) .This model can be used for part 3 of the project .This model is trained on a pretrained model of MobileV2.net with weights taken from imagenet.

**PS:Aniket you can add the reference to the data-set you used for the third model.Also if you find any difference in specifications of the model here with the ones mentioned in the program do update it here .**

**Running the Code**

1. Before running the code make sure you have python , anaconda virtual environment and pip installed and of course a text editor like sublime text or vscode or any ide.
2. Store all the contents of code together in the same folder.
3. Make sure to add the index.html file in a templates folder and add an empty folder named ‘static’.