

# FONTS

## ADDING A NEW DIMENSION TO A DOCUMENT

Beth is a part of the graphics designer team in a Mass Customization Company and works on enhancing the customer's artworks like Business cards, Postcards etc. He needs to choose from a variety of fonts for a document based on the context. Many times, he requires to replicate the font used in the customer's uploaded image for adding new text. Estimating the font type manually from the image is a troublesome process. We want to build a Computer Vision-based solution for easing Beth's work.

## FONT MATCHING

You need to classify the fonts used for the written text in a given image. Write a software that will take an image as input and provides recommendations of matching fonts with the confidence level.

You can assume the following while developing your solution:

- All the images will have known text "Hello, World!"
- All the images will have white background with black text.
- Sample input images (<https://cimpressexpress.com/v/imagicasamples>)
- The number of fonts you need to consider are restricted to the fonts list provided below

```
https://fonts.google.com/specimen/Oswald
https://fonts.google.com/specimen/Roboto
https://fonts.google.com/specimen/Open+Sans
https://fonts.google.com/specimen/Ubuntu
https://fonts.google.com/specimen/PT+Serif
https://fonts.google.com/specimen/Dancing+Script
https://fonts.google.com/specimen/Fredoka+One
https://fonts.google.com/specimen/Arimo
https://fonts.google.com/specimen/Noto+Sans
https://fonts.google.com/specimen/Patua+One
```

*Note:* There can be known solutions for font classification available externally, but we don't you want to use them, e.g. (<https://www.myfonts.com/WhatTheFont/>). **We want you to generate your own training data and build the model. You have to use Python as the programming language.**

## BOUNDING BOX OVER THE "HELLO WORLD!"

There can be multiple "Hello World!" written in an image with different fonts. You need to extract each one of the "Hello World!" instances and classify its font. The output should be the bounding boxes coordinates containing the "Hello World!" and the type of font with confidence score as mentioned below.

```

Output: {"detectedFonts": [
    {
      "boundingBox": {
        "x": int,
        "y": int,
        "width": int,
        "height": int
      },
      "font": "string",
      "confidence": float
    }
  ]
}
Input: {"url": image url}

```

\* "detectedFonts" should be an array containing information about all the detected "Hello, World!" in an image and classified font for it.

## Your submission should include

1. Source code with instructions on how to run it.
2. Data Creation Script.
3. Data used for training.
4. Training Script (Deep learning)
5. Model Architecture File.
6. Model File with trained weights
7. Summary of your final findings.
8. Your software should be able to add more fonts if needed.

Words can be of different sizes (sample image for testing):

Hello, World!      Hello, World!  
**HELLO, WORLD!**  
 HELLO, WORLD! *Hello, World!*  
**Hello, World!** Hello, World!  
 Hello, World! Hello, World!