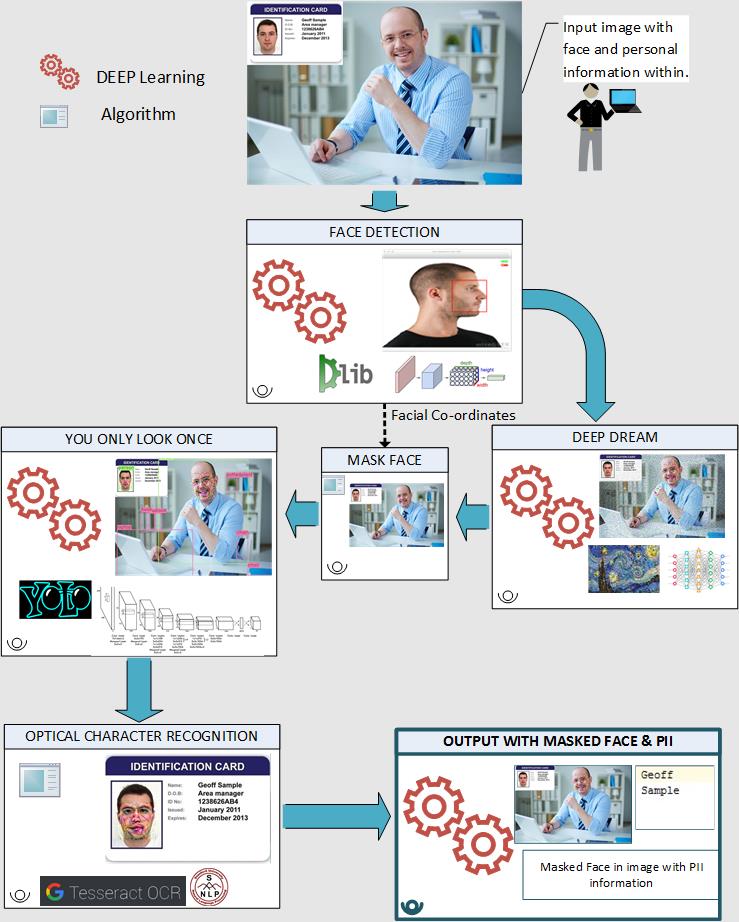
Detect & Mask PII Within Image For Compliance Compliance

Information Technology Solutions



## **Face Recognition API**

Face is detected using face\_recognition package in Python, which mainly uses DLib as a reference library. Image, then, is obfuscated using Google’s Deep Dream API. It first loads an image into a numpy array, and then finds the face encodings. These encodings can be compared with other images to see whether image is matched or not. Face encodings return a 128 dimension face encoding of each face in an image.

## **Process Workflow:**

Text in that image is detected with Optical Character Recognition (OCR), and person’s names are identified by implementing Stanford NLP’s Named Entity Recognition (NER).

GDPR obliges the companies under EU Governance to protect their employees’ personal data under any security breach and failing to do so, companies would have to face penalties and other consequences.

One of the challeges in identifying personal information is when images of individuals are contained in a document or when personal information (like names) is within the image. Our approach is to make a system that can accurately classify this personal information which is present in a document without human intervention. To do so, we employed the use of various Deep learning concepts.

Scope of this work is to detect:

* Face of individual within image
* Classify other objects shown within image
* Personal information like First Name & Last name

This system can be further fine tuned to be more elaborative.

In this approach, we try to detect personal information in an image by detecting human faces and recognizing names inside it. We start by detecting human faces in an image, which is obfuscated using Google’s Deep Dream API. Other objects in that image are, then, identified using YOLO.

Abstract

General Data Protection Regulation (GDPR) states that all the companies must provide a reasonable data protection for personal data of their customers. Personal Data includes any information relating to an individual, of private, professional or public life. It can be anything from a name, a home address, a photo, an email address, bank details, medical information, or a computer's [IP address](https://en.wikipedia.org/wiki/IP_address). In this approach, we cater to design a system which can detect and/or mask personal information of an individual like First Name & Last Name, and facial image through which individual can be identified. Addtionally this system also detects other objects within the image.

We have detected face in given images and obfuscated them by using Google’s Deep Dream algorithm, objects within the same image are detected using YOLO algorithm & Textual information using Stanford NLP’s Named Entity Recognition (NER).

Acronyms: Personally Identifiable Information (PII), Convolutional Neural Networks (CNN), Deep Dream, Named Entity Recognition (NER), Natural Language Processing (NLP), You Only Look Once (YOLO).

Prepared By : Neel Parikh, Tirth Thakkar

Guidance By: Dhaval Mandalia

## **Convolutional Neural Network(CNN)**

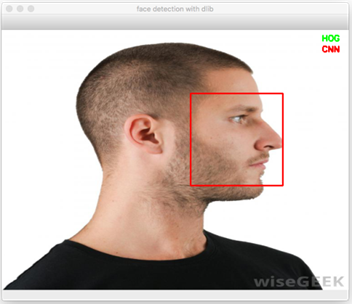
Convolutional Neural Network(CNN) is one of the main categories of image classification and recognition. CNN mainly trains a model from a set of input data that is used for detecting a specific entity in an image. CNN has different layers, like convolution, ReLU, max pooling etc. Convolution layer extracts features from an input image. ReLU basically introduces non-linearity in the network. Pooling is used to reduce the size of an image so that operations can be performed faster. . A sigmoid function, for example, would have the following equation:

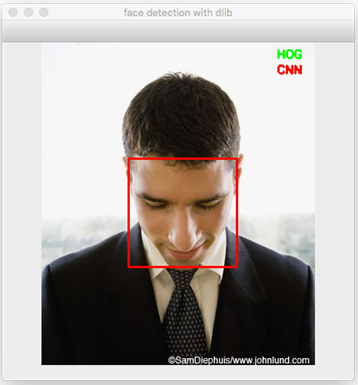
S(x) = 1/(1+) (1)

A simple CNN process would look like following:

## **DLib**

DLib is a C++ based toolkit that contains machine learning algorithms to solve real world problems. The frontal face detector in dlib is based on histogram of oriented gradients(HOG) and linear SVM. DLib also provides CNN based face detector which can detect faces almost in all angles. While the hog based face detector takes less time, CNN based face detector would take considerably more time. However, the results show much more accuracy for CNN based face detector.

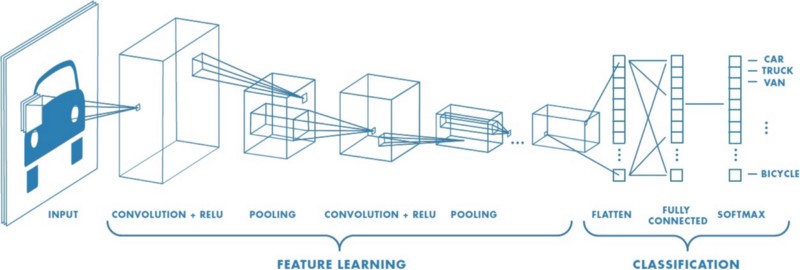




## **D. Deep Dream**

Deep dream is a Google API that uses CNN to find and enhance patterns in images and create psychedelic image from that. Deep Dream mainly reverses its neural network. When any random image is given and told to find an object that is not present in an image, rather than failing to find the object, the system starts to recreate image in a way that it finds the object it was supposed to identify. Thus, it creates a hallucinating image. In practical implementation, lower level layers identifies the edges of the object, and as layers are increased, patterns are created into an object shape which are overlapped repeatedly to create a hallucinating image.

*Fig.2* *Convolutional Neural Networks*

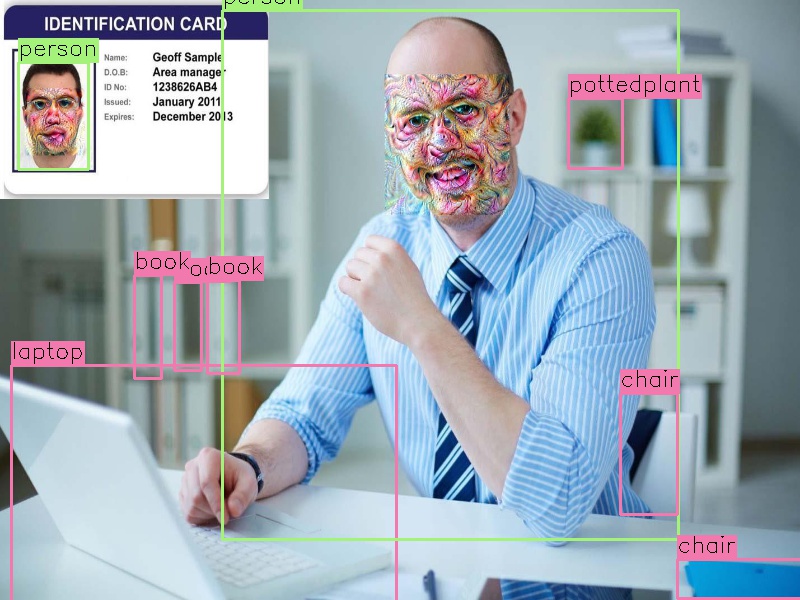


*Fig 1. Face detection*

## **Object Detection and recognition.**

In this section, we would identify all the different objects present in the obfuscated image by the application of YOLO algorithm. This section identifies 80 categories of images. YOLO, the name stands for “you look only once”. The algorithm detects various objects by just scanning the image once, rather than the traditional sliding window techniques. This reduces the response time to a great extent. Yolo divides the image in to S\*S grid and each grid cell predicts the bounding box along which an image might be present. The prediction of these grids are later done by Convolutional neural network(CNN). CNN considers only those grids which have confidence score above some predefined threshold. On normal scenario, object detection would look like following:





The model used in this process uses Octave=2 and Iteration = 100. **They can be further scaled up as per requirement to give more obfuscation** . An image before and after applying deep dream would look like following:







*Fig 3a. Original Image*

*Fig 4a. Input Image*

*Fig 3b. Image After Deep Dream Application*

*Fig 4b. Image With Objects Detected*

*Fig 3c. Image With Masked Face*

*Fig 5. Face Obfuscation using Deep Dream.*

*Fig 7. Object Detection by YOLO*

*Fig 5b. Named Entity Recognition from OCR*

*Fig 5a. Input Image*

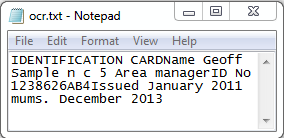
*Fig 5b. Optical Character Recognition.*

**References:**

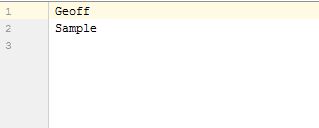
* About DLIB: <http://dlib.net/>
* Deep Dream Functions: <https://en.wikipedia.org/wiki/DeepDream>
* Deep Dream Architecture: <https://github.com/google/deepdream>
* YOLO: <https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>
* Tessaract OCR: <https://github.com/tesseract-ocr/>
* Named Entity Recognition: <https://nlp.stanford.edu/software/CRF-NER.shtml>

## **Natural Language Processing**

In this phase, we used the concept of natural language processing to identify the private information such as first name and last name. Natural Language processing is a technique which can analyze large amount of human language and can identify the semantics present in the language. We used Stanford NLP to identify the various private information from the text. We used Stanford Named Entity Recognition to identify the first name and last name. Stanford Named Entity Recognition labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. Actual implementation is shown in an image as below:



First name and Last Name extracted from OCR using Named Entity Recognition.



## **Optical Character Recognition(OCR)**

In this phase, we applied optical character recognition to identify the characters embedded into an image. Optical character recognition is a technique which transforms mechanical or electrically typed texts into machine encoded texts. In OCR the image is scanned for light and dark areas in order to identify each alphabetic letter or numeric digit. We used tesseract OCR, which is the open source API developed by Hewlett Packard labs, licensed under Apache and has been sponsored by Google since 2006. Applying OCR on an image would like as following: 