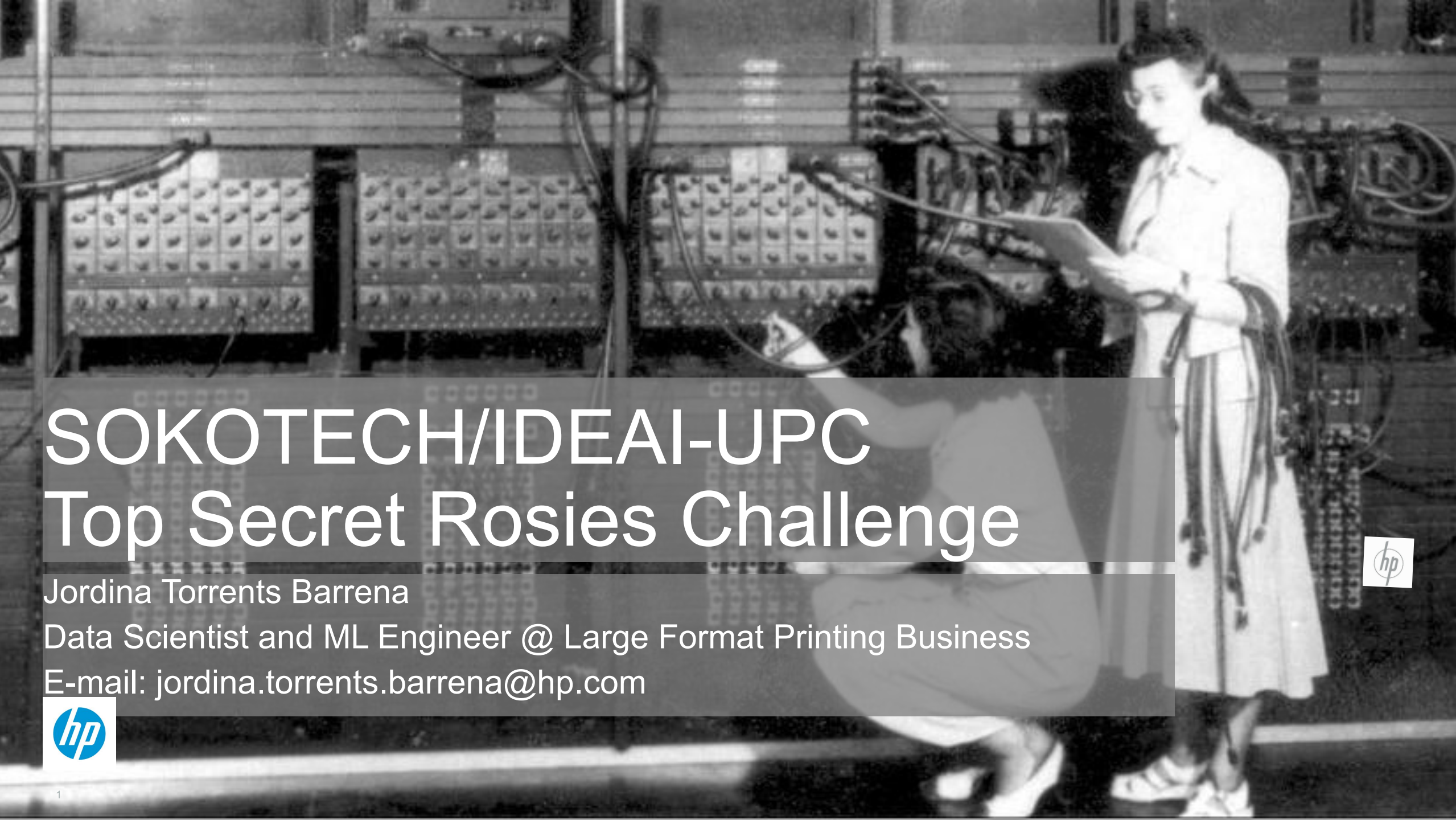


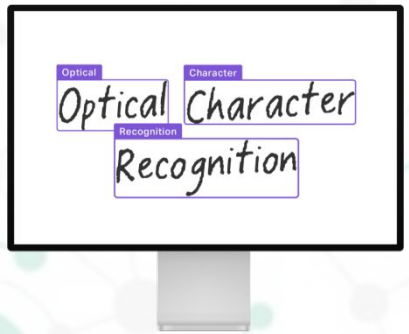
# SOKOTECH/IDEAI-UPC
# Top Secret Rosies Challenge

Jordina Torrents Barrena
Data Scientist and ML Engineer @ Large Format Printing Business
E-mail: jordina.torrents.barrena@hp.com

# Have you ever wondered how difficult is to deal with all these stuff for vision impaired people?

# We think technology can help them in their day to day, but how?

## Combining leading edge technologies for helping people – Connect the dots




### 1. Classifying different pictures

Areas of potential interest:
- Image Categorization
- Content Understanding
- Tagging (or Indexing)

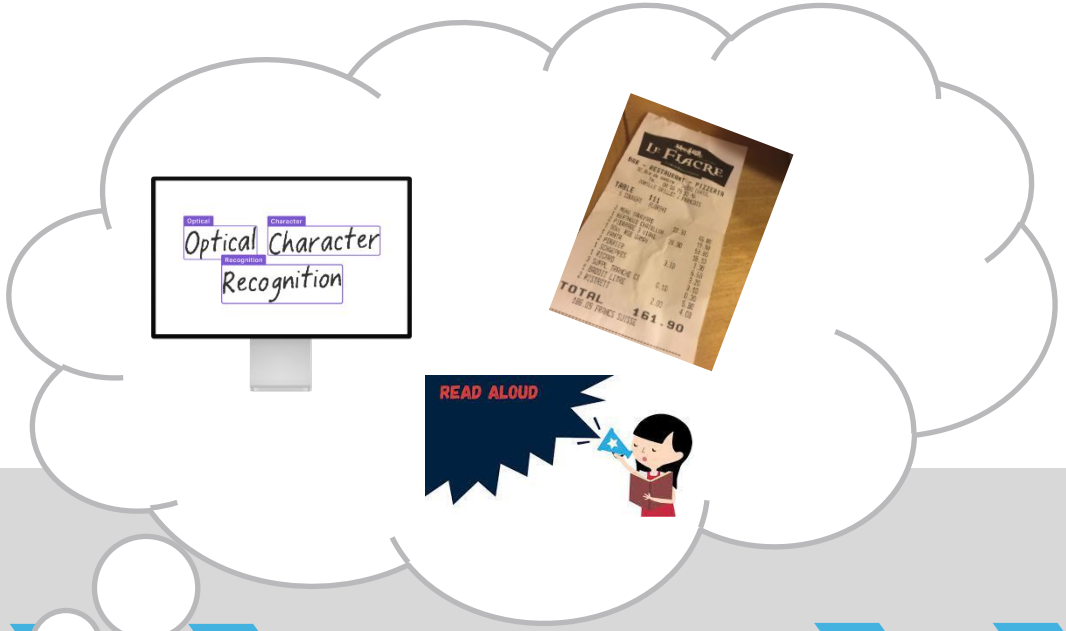
### 2. Recognizing text in the images

Areas of potential interest:
- Text Detection
- Text Recognition
- Text Correction
- Text Summarization

### 3. Reading out loud through text-to-speech

Languages of interest:
- English

# Imagine how it Works…




1


2

3

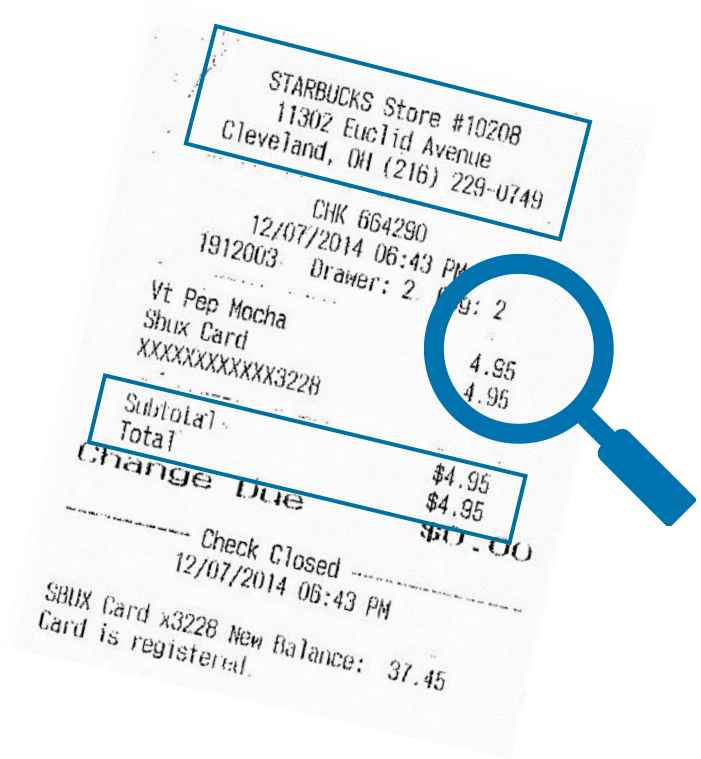
# Challenge Proposed Steps

| | 1. Classifying different pictures (8-10 days) | 2. Recognizing text in the images (8-10 days) | 3. Reading out loud through text to speech (3 days) |
|---|---|---|---|
| **Goals** | To measure the categorization between tickets, floorplans and documents.<br>Image Classification:<br>• Confusion Matrix<br>• Accuracy = 0.90 | Text Detection:<br>• Mean Average Precision = 0.80<br>Text Recognition and Correction:<br>• Levenshtein (CER) = 0.80, (WER) = 0.85<br>Execution time <= 0.1s | To turn recognized and summarized texts into natural-sounding speech.<br><br>The text read should be in English. |
| **Tools** | Image Classification to explore:<br>• Sklearn: Feature Extraction + Traditional Machine Learning<br>• Pytorch: Convolutional Neural Networks + Transfer Learning (*e.g.*, ImageNet) | OCRs, Summarization & Spell Checkers to explore:<br>• Tesseract 4 (LSTM)<br>• EasyOCR<br>• Google Cloud Vision API<br>• Regular Expressions<br>• TextBlob, JamSpell, NLTK, Spacy | Text-to-Speech libraries to explore:<br>• Google Text-To-Speech<br>• Amazon Polly<br>• gTTS 2.2.2<br>• pyttsx3 2.90 |
| **Deliverables** | Github Code (master) and Pages (report)<br>Executable:<br>• Input: image (.png, .jpg)<br>• Output: caption (.txt) | Github Code (master) and Pages (report)<br>Executable:<br>• Input: image (.png, .jpg) with text | Github Code (master) and Pages (report)<br>Executable:<br>• Input: text or caption (.txt)<br>• Output: audio (.mp3) |

1 2 3

Week1 Week2 Week3 Week4

# 2. Recognizing text in the images




1. Text Detection (bboxs) using Tesseract.
2. Text Recognition (from tickets to .txt files) using Tesseract.
3. *Date*, *total*, *company* and *address* identification from the ticket.
4. Text summarization using regular expressions.

## Tesseract

Tesseract was originally developed at Hewlett-Packard Laboratories Bristol and at Hewlett-Packard Co, Greeley Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and some C++izing in 1998. In 2005 Tesseract was open sourced by HP.
The latest (LSTM based) stable version is 4.1.1, released on December 26, 2019.

### Installation

`pip install pytesseract` (python version)
`sudo apt install tesseract-ocr` (original version)

### Github

The public repository with the original source code can be found here.

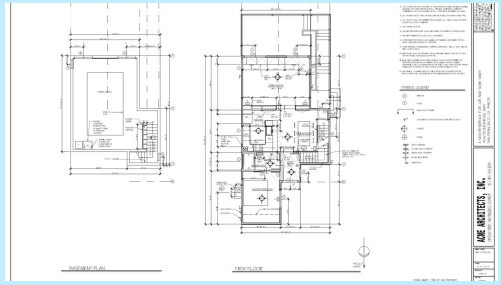
### Documentation

The user manual can be found here. Tesseract can be used with CLI or native code.

# Databases you will use…

Download data here!

## Classifying different pictures



The Royal Borough of Kensington and Chelsea

ICDAR-2019-SROIE

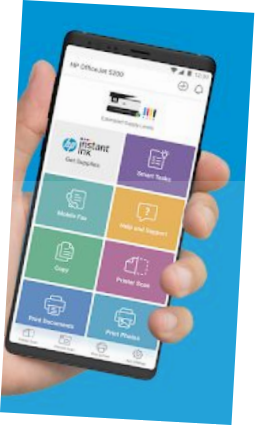
SROIE

RIMES and IAM Handwriting

## Recognizing text in the images

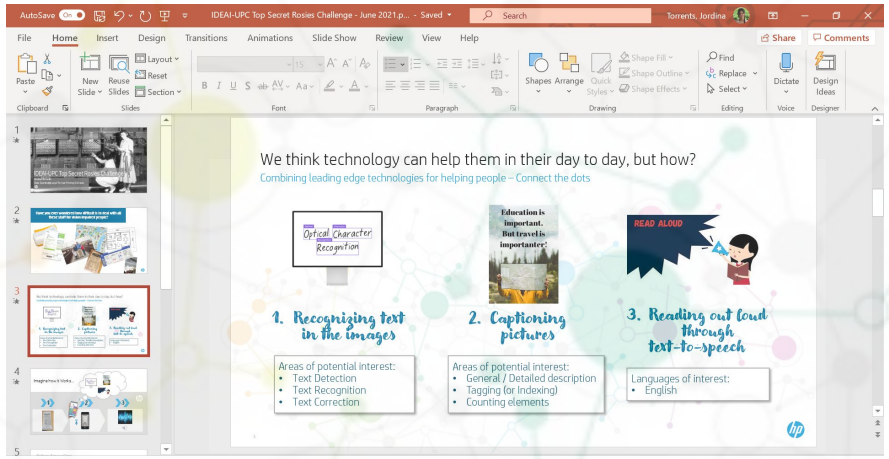
ICDAR-2019-SROIE

SROIE

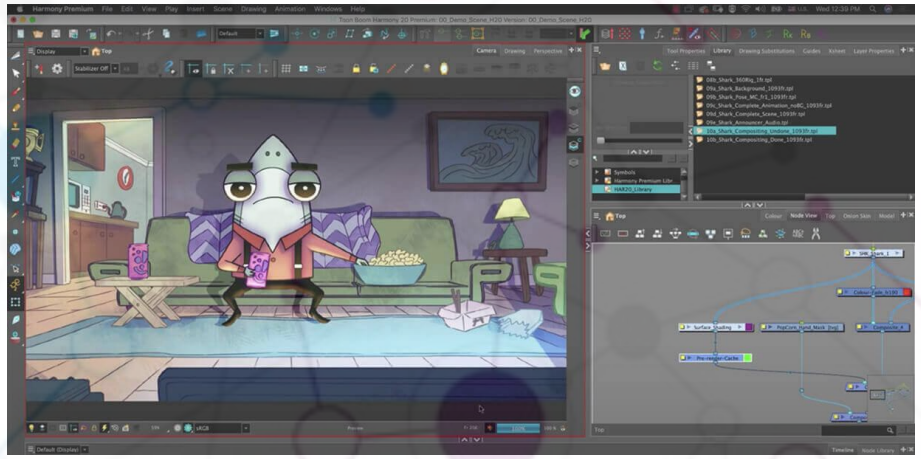


# Marketing your HP App!

## Presentation Deck




## DEMO



**Aspects to cover** (10 min + 5 min QA):

- Pipeline presentation.
- Justification for tool selection (benchmarking).
- Averaged results on the full database.
- Computational cost (time + resources).
- Strengths and limitations of the pipeline.

**Aspects to cover** (5 min):

- Real-time execution of each phase.
- Trial with 5-10 already tested images.
- and … bunch of data surprise! ;)


keep reinventing