

# Project: Second Sight through Machine Learning

## Final Presentation

Lisa Wu and Camille Church  
W207 Machine Learning  
Fall 2022

**Berkeley**  
UNIVERSITY OF CALIFORNIA

# Section one: Motivation and Question Definition

# Background Information

27%

of blind people live  
below  
the **poverty** line

\$38k

Median household  
income

1.3 million

people in the US are  
legally blind.

Same number of  
people as the  
population of **Maine**.



## OrCam MyEye Pro

Starting at **\$3,999**

*“Speaks any **readable** text aloud on-demand”*

## Actual Reviews

*“What really gets me about this is that we had to fund raise in order to be able to afford this. And what is it really? It’s nothing more than a \$4000 bill (money) identifier. Totally feel cheated by this company and this product.”*

*“this is so expensive, i have been thinking to buy it for a family member who is in big need to be independent but it is really unaffordable”*

# Objective

## Fall 2022

---

- ❖ Proof of Concept in Notebook
- ❖ Create a model(s) that can accurately recognize handwritten text images
- ❖ Model should correct for misspelled words

## Capstone and Beyond

---

- ❖ Mobile Application
- ❖ Identifies handwritten, currency, typed text, and computer screen text.
- ❖ All compute must be performed on device.
- ❖ 100% free for download and use

# Data and Model Pipeline

## Data Processing

- ❖ Train/Validation/Test Split
- ❖ Exploratory Data Analysis
- ❖ Image and Label Preprocessing
  - Data augmentation
  - Vectorize labels

## Computer Vision

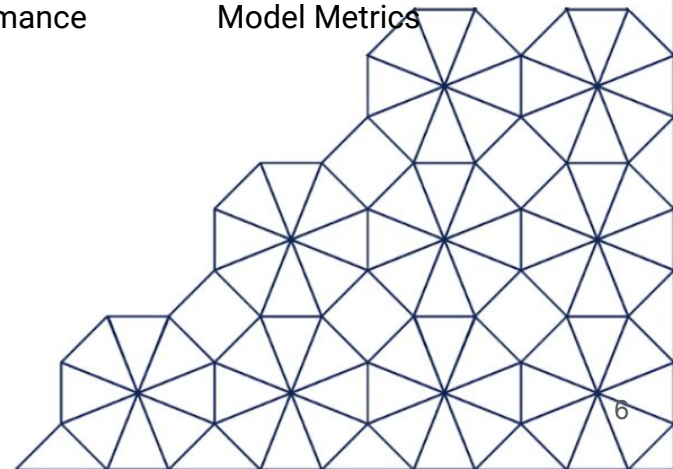
- ❖ Explore standard NN models
- ❖ Explore CNN and RNN models
- ❖ Design and implement evaluation metrics

## NLP

- ❖ Spell check output words to account for misspelled words
- ❖ Evaluate performance metrics f

## Output

- ❖ Prediction for each word image
- ❖ Evaluate and Compare Model Metrics





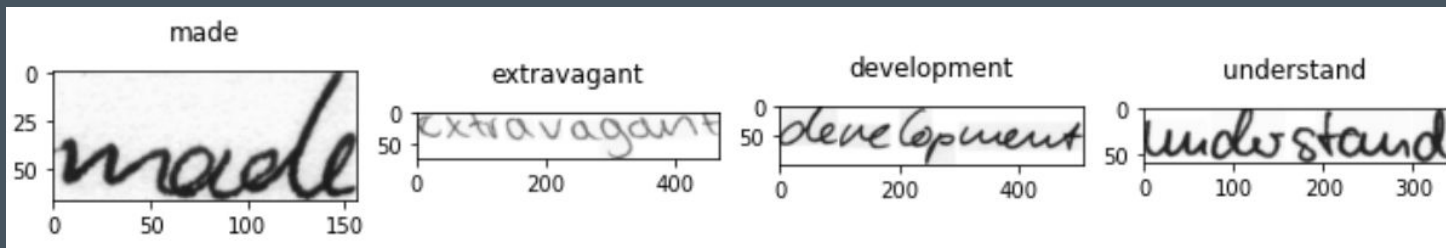
# Section two:

# Data Exploration

# Data Characteristics

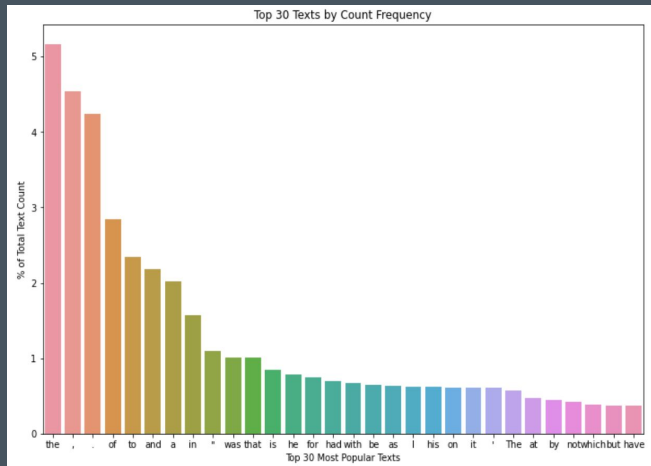
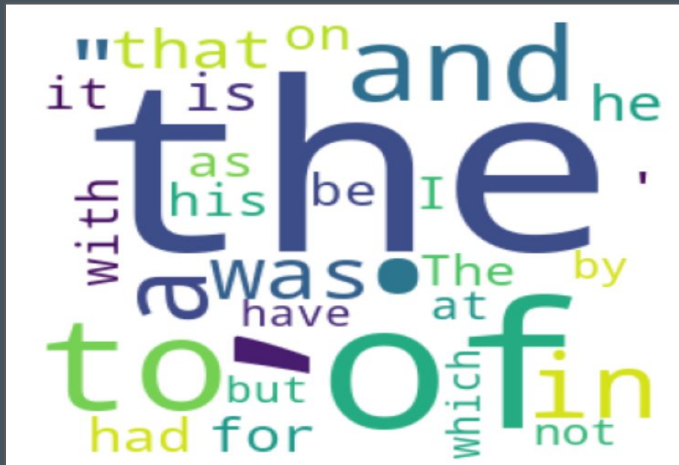
Type	Handwritten text images
# of Images	96,456 words in English
# of Unique Texts	11,528
# of Unique Characters	78
Image Size	Varies
Length of Text	1 - 21 characters

## 4 Examples of Text Labels





# Top 15 Most Popular Texts



Interestingly,

- The most popular text is “the”, which represents ~5% of the dataset
- The top 15 texts account for ~31% of the total text dataset

# Key challenges to model handwritten text images

- Unstructured data
- High variance in style and quality given the handwriting nature
- Multiple output (multiple characters) of each label
- No fixed length of strings in each text
- Order matters for sequential data
  - Recognizing the correct characters but in the wrong order is not enough
- Duplicate strings (e.g. too) could be valid

# Key considerations of Labels

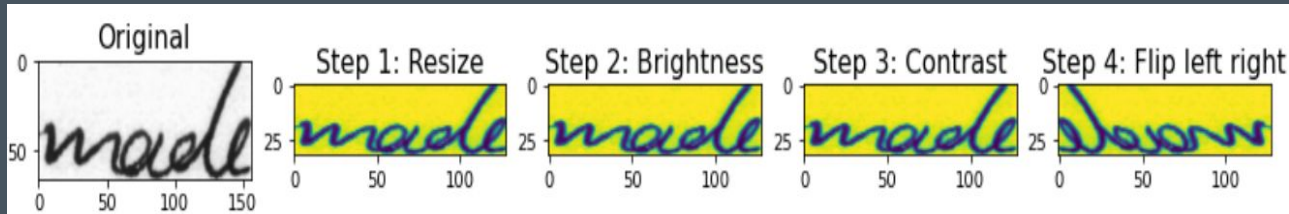
## Two options:

- **Use the entire text as the label**
  - **Pros:** Little transformation work
  - **Cons:**
    - Requires a vast amount of data to train each text label
    - The model may not generalize well (~1M words in the English vocabulary)
- **Use the single characters in each text as multiple labels**
  - **Pros:** Only 78 common characters in the dataset, more efficient and generalizable
  - **Cons:** Higher complexity, handle timing sequence and minimize false positives (e.g. valid duplicate characters in a text)

↑  
Our planned approach

# Key steps of Data Preprocessing

- Read and encode input images
- Standardize the image size and normalize pixel values
- Augment the data (e.g. rotation, brightness, contrast, flip)



- Encode characters to numbers and inverse back to characters
- Split data into train, validation and test dataset

# Section three: Modeling Approach and Evaluation Metrics

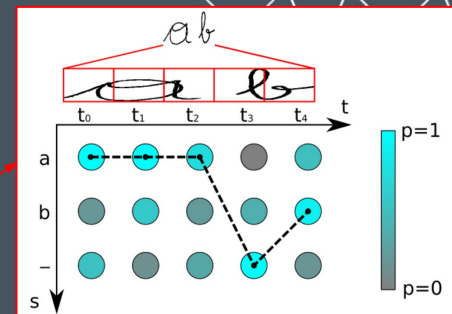
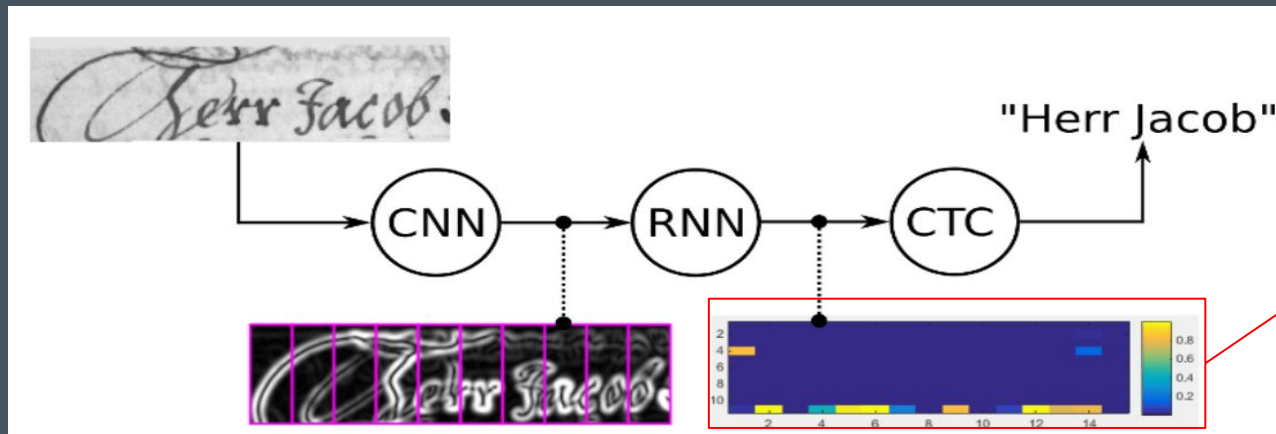
# Our Model Experiments

<b>Baseline Model</b>	CNN + CTC Loss/Decoder
<b>Enhanced Model</b>	CNN + RNN (LSTM) + CTC
<b>Target Model</b>	CNN + RNN + CTC + NLP Spell Check
<b>Further Exploration</b>	Transformer

Note: we experimented with shallow neural network (e.g. multi-classification) and standard MLP neural network models with categorical cross-entropy loss function. These models failed to capture the complex features of the dataset and produced less than 10% accuracy



# How does the core model work for text images?



- **Convolutional NN(CNN):** Convolutional layers extract a sequence of features
- **Recurrent NN(RNN):** Propagate information through the sequence and output character-scores for each sequence-element (matrix)
- **Connectionist Temporal Classification (CTC):** calculate the loss to train the NN and decode the matrix to get the text contained in the input image

# Models Evaluation Metrics

## ❖ Measure Accuracy by individual letter

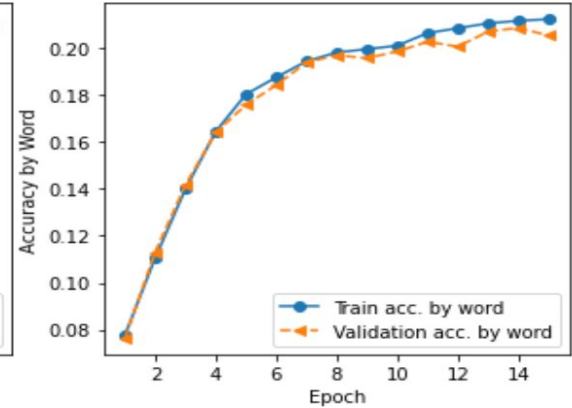
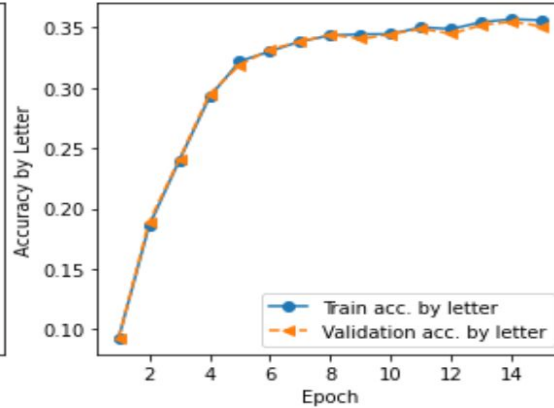
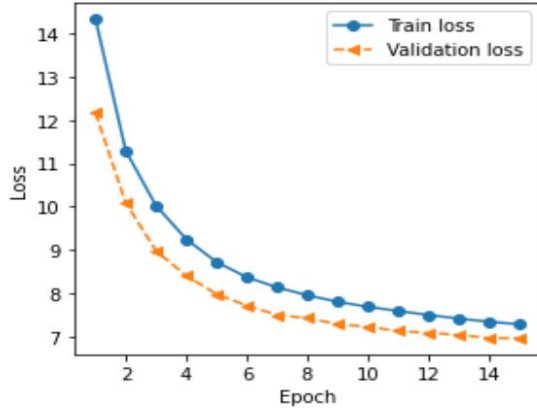
$$\text{Accuracy by Letter} = \frac{\text{Count of Accurate Letters}}{\text{Total Count of Letters}}$$

## ❖ Measure Accuracy by each word

$$\text{Accuracy by Word} = \frac{\text{Count of Fully Matched Words}}{\text{Total Count of Words}}$$

# Section four: Model Performance Comparison

# Baseline Model (CNN + CTC Loss) Performance



## Evaluation Metrics

Dataset	Accuracy by Letter	Accuracy by Word
Training (86,810)	35.6%	21.3%
Validation (4,823)	35.1%	20.6%
Test (4,823)	<b>34.1%</b>	<b>19.6%</b>

## Observation

- ❖ CNN does a better job than the standard NN for this dataset
- ❖ Loss curve declines and accuracies improve nicely for 15 epochs
- ❖ However, CNN is not sufficient for sequential data

# Enhanced Model (CNN + RNN+ CTC Loss)

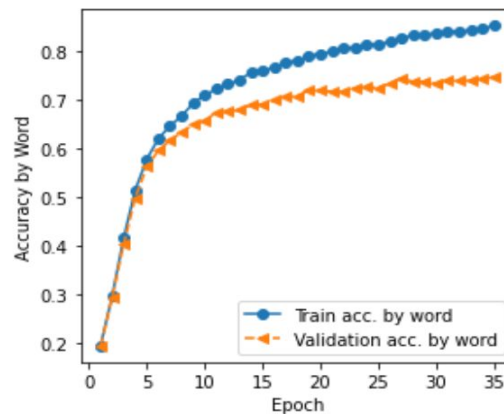
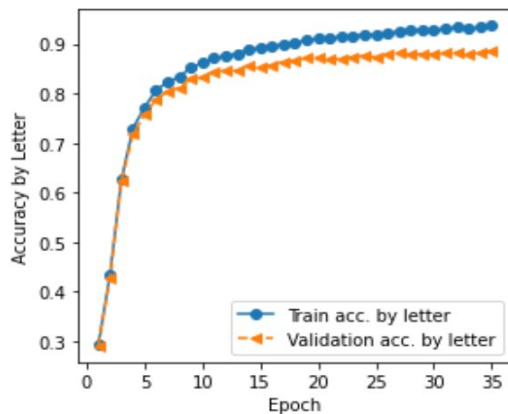
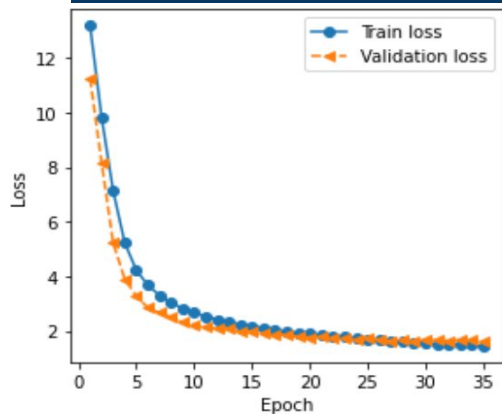
Much improved model performance than the Base Model!!

	Train Letter Accuracy	Val Letter Accuracy	Test Letter Accuracy	Train Word Accuracy	Val Word Accuracy	Test Word Accuracy	Kernel Size	Optimizer	Drop Out	RNN/LSTM	NLP Spell Check	Epochs
<b>Base</b>	0.356	0.351	0.341	0.213	0.206	0.196	3,3	Adam	0.2			15
	0.234	0.233	0.227	0.130	0.125	0.124	2,2	Adam	0.2			15
	0.332	0.325	0.322	0.197	0.192	0.187	3,3	SGD	0.2			15
<b>Enhanced</b>	0.875	0.850	0.842	0.724	0.676	0.667	3,3	Adam	0.2	[128,64]		15
	0.888	0.844	0.843	0.756	0.684	0.685	2,2	Adam	0.2	[200,128,64]		15
	0.903	0.860	0.864	0.778	0.697	0.701	3,3	Adam	0.2	[200,128,64]		15
	0.919	0.872	0.866	0.813	0.718	0.711	2,2	Adam	0.25	[200,128,64]		35
	0.937	0.886	0.885	0.852	0.746	0.751	3,3	Adam	0.25	[200,128,64]		35

Final version of the Enhanced Model

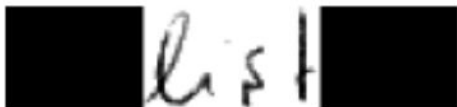
# Enhanced Model (CNN + RNN + CTC Loss)

## Final Training Results

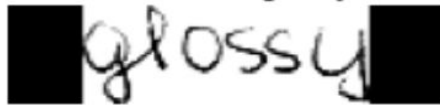


## Sample Predictions

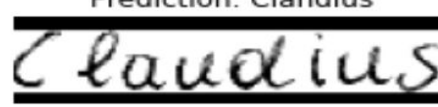
Prediction: list



Prediction: glossy



Prediction: Clandius





# Target Model (CNN + RNN + CTC Loss + NLP Spell Check)

Apply NLP Spell Check further improved predictions on test data!

	Base Model	Enhanced Model	Target Model
Accuracy by Letter	34.1%	88.5%	89.0%
Accuracy by Word	19.6%	75.1%	81.3%

NLP Spell Check Helped!

Label: Claudius

Prediction: Clandius  
Spell Pred: Claudius



NLP Spell Check Didn't Help

Label: returned

Prediction: retumed  
Spell Pred: resumed



# Explore Transformer Models

## ❖ Overcome RNN limitations

- Remove RNN layers
- Parallel processing to speed up calculations
- Handle long sequences efficiently

## ❖ Potential to Enhance Model Performance

**NLP Spell Check Didn't Help**

Label: returned

Prediction: returmed  
Spell Pred: resumed



returned

**Transformer Model did it!**

Label: returned

Transform Pred: returned



returned

# Out-of-Box Transformer Model Performance

- ❖ **Explored Microsoft TrOCR encoder and decoder models**

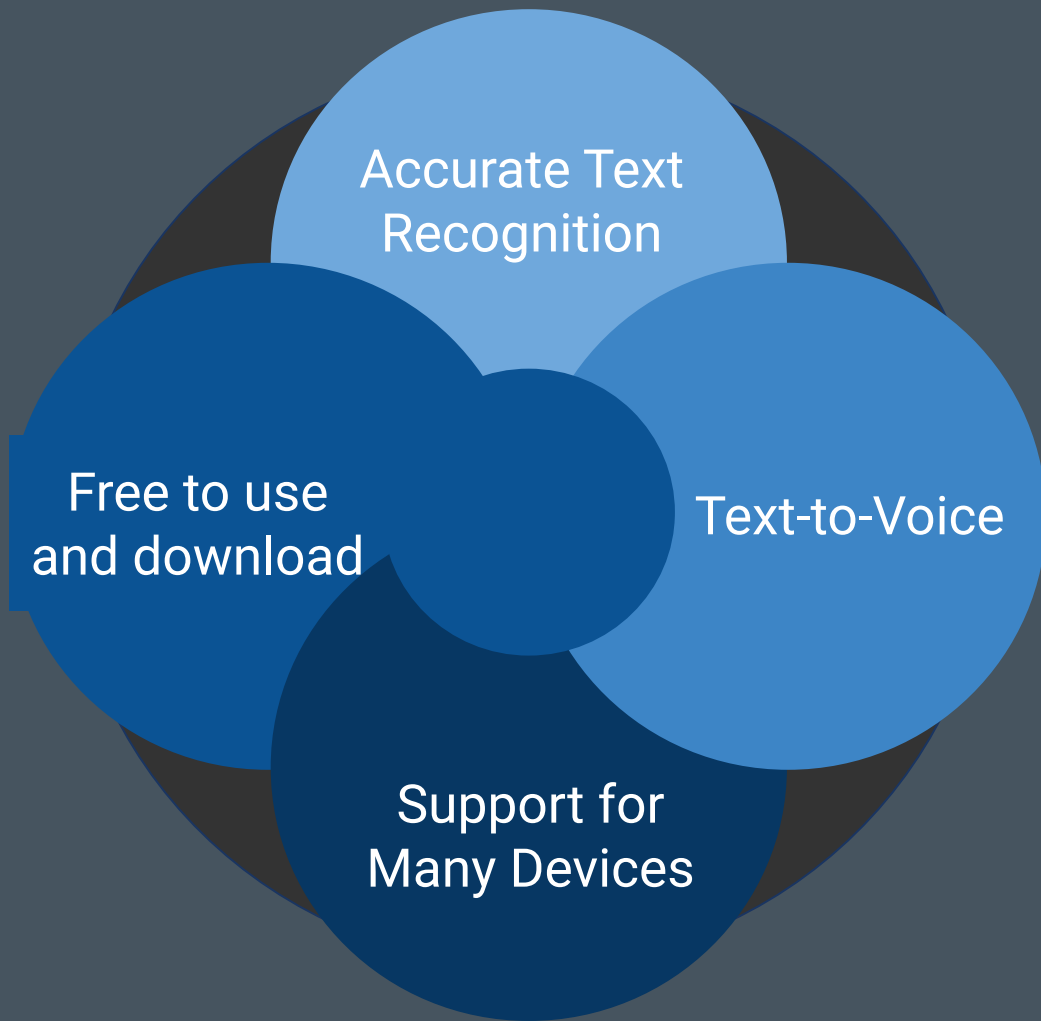
- Pre-trained on IAM and need further refining on this dataset

- ❖ **Model Performance Comparison**

Test Data	Enhanced Model	“Out of Box” Transformer Model
Model Train Time	2-3 weeks	8-10 hours
Accuracy by Letter	89%	60%
Accuracy by Word	75%	41%

We plan to refine the Transformer Models performance in our Capstone project

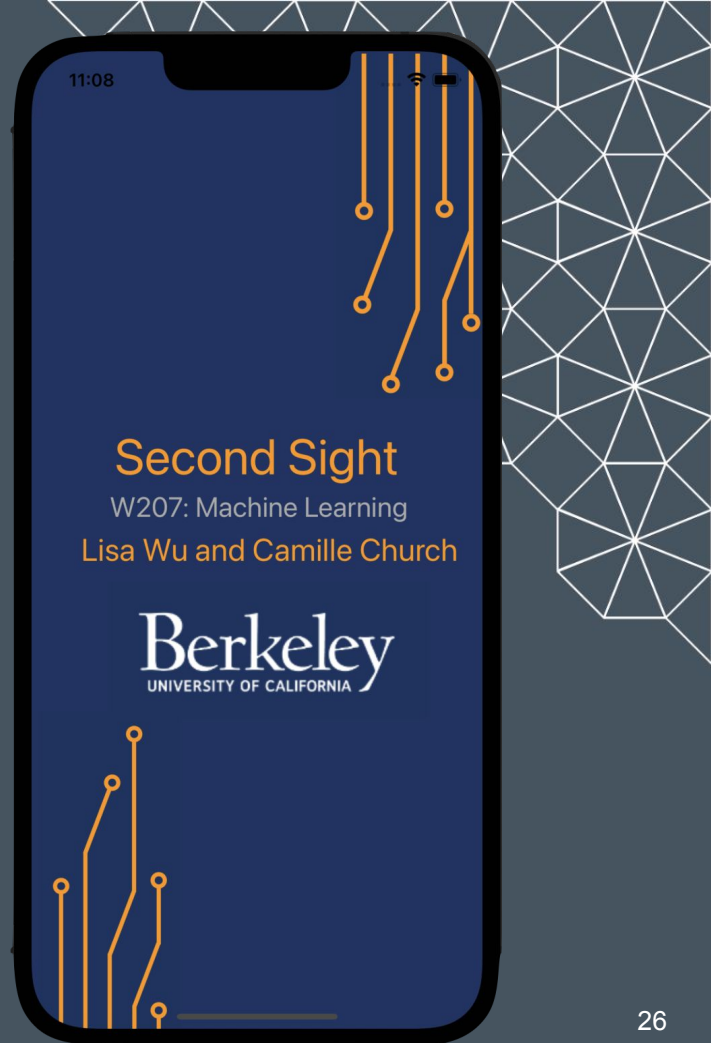
# Section Four: Future Work



## Mobile App Requirements

## Current Limitations

- ❖ Only available on iPhone
  - Newer phones
- ❖ Doesn't implement NLP modeling
- ❖ Doesn't implement same photo preprocessing as notebook implementation





## Mobile App Demo

10:57



Dear Dad,  
Happy Birthday!  
Love  
CJ

Predicted Text:

Dear Dad, Happy Birthday! LOVe CJ

# Questions?

\*\*\*\*NOTE: If you have an iPhone and would like to be a tester of this app, or know anyone that could benefit from it, please let us know and we will add you to the beta release.

Thanks! Lisa and Camille

## References

- [Connectionist Temporal Classification](#)
- [Handwriting Recognition using Machine Learning](#)
- [Build a Handwritten Text Recognition System](#)
- [Microsoft TrOCR model](#)
- [OrCam MyEye Pro - The Most Advanced Wearable Assistive Device for the Blind and Visually Impaired.](#)
- [Blindness Statistics](#)
- Python Machine Learning, by Raschka and Mirjalili
- Transformers for NLP, Second Edition, by Denis Rothman

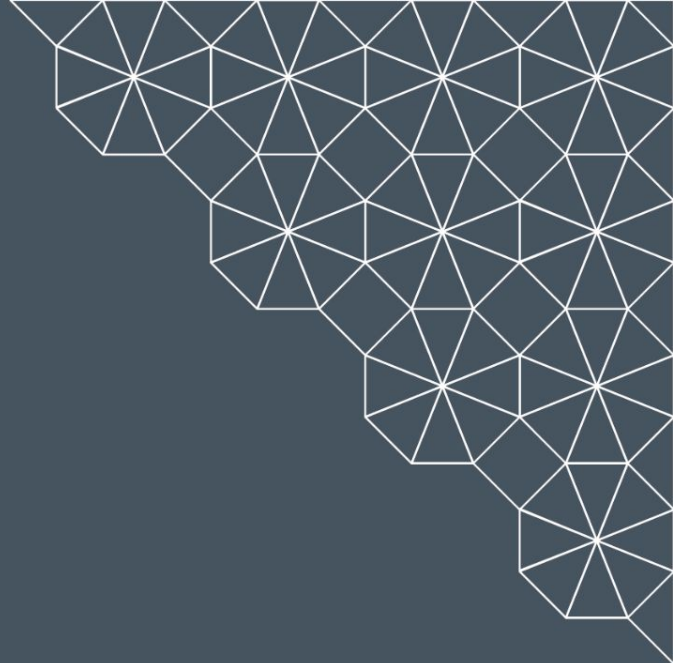
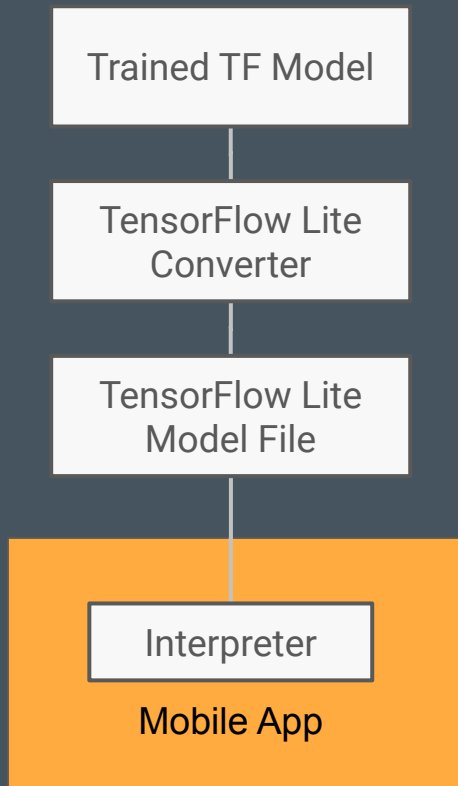
## Github Repository

- [https://github.com/Camille2985/w207\\_team\\_project](https://github.com/Camille2985/w207_team_project)

# Contributions


- ❖ **Camille Church:** Team discussions, dataset selection, project vision, model theory research, accuracy metrics function, model experiments (Optuna, CNN, RNN, NLP), mobile app POC, and presentation slides
- ❖ **Lisa Wu:** Team discussions, EDA, model theory research, refine accuracy metrics and graphs, model experiments (standard NN, CNN, RNN, NLP, Transformer), and presentation slides

# Mobile App Architecture (iPhone)



# Real World Examples


9:35



Predicted Text:

iHANOVER CHICK PEAS Garbanzo Brans VEGAN  
GLUTEN FREE NET WT 7.73 OZ (2209)


9:34



Predicted Text:

LAUNDRY

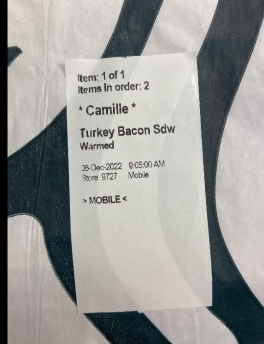
9:37



Predicted Text:

Walgreens Pain NOC 0363-051012 Reliever  
ACETAMINOPHEN 500 mg/ PAIN RELIEVER / F...

9:39



Predicted Text:

item: 1 of 1 Items in order: 2 \* Camille \*  
Turkey Bacon Sdw Warmed 06-Dec 2022 9:05:00 AM...