Aakash Madabhushi
A20416408

Prince Hodonou
A20371916

Description of the Problem

In modern software engineering, applications are getting more and more complex. Client demands to build and deploy apps due quickly are increasing tremendously. The process of creating applications can significantly speed up if user interface development time decreases.

Therefore we propose a way to allow developers to decrease development time by turning their user interface sketches into code by creating a deep learning model that can use handwriting recognition to detect UI elements drawn from the sketches. Though there are typical drag and drop UI platforms out there, such as WIX and Weebly, these are not robust to create various apps. What is needed is a developer tool that gives developers a stepping stool to create robust applications quickly.

Method Planning to apply

We plan to use similar methods as in our research paper. In the research paper, a network built was to identify App Inventor UI elements from screenshots/sketches. We plan to use similar methods to identify Flutter UI elements.

We plan to implement a Convolutional Neural Network model that can recognize UI elements' sketches and classify them to Flutter UI components. We plan to use the YOLOv4 model for detecting objects in images by adopting a transfer learning strategy from a pre-trained YOLO network with the ImageNet dataset. The model should also be able to localize each component in comparison to the other components. We then generate UI code based on the classification and location of each element.

Data Planning to use

We plan to create our data set of hand-drawn UI elements in order to train the model. Because labeling each component in a sketch may become too expensive and time-consuming, we may use weakly supervised learning. We can use Multiple Instance Learning to allow the learning algorithm to learn with weakly labeled data.

Multiple Instance Learning is a Weakly Supervised Object Localization that allows a learning algorithm to learn with weakly labeled data. Unlike typical classification methods, in multiple instance learning, each instance is not individually labeled. Instead, the algorithm provides a set of "bags" that are labeled positive or negative. Each bag contains many instances. Essentially, a bag is labeled negative if all the instances in it are negative. If there exists a positive instance in the bag, it is labeled positive (See figure 1). Now the learner's goal is to produce a concept that will label individual instances correctly in the future.

Aakash Madabhushi
A20416408

Prince Hodonou
A20371916

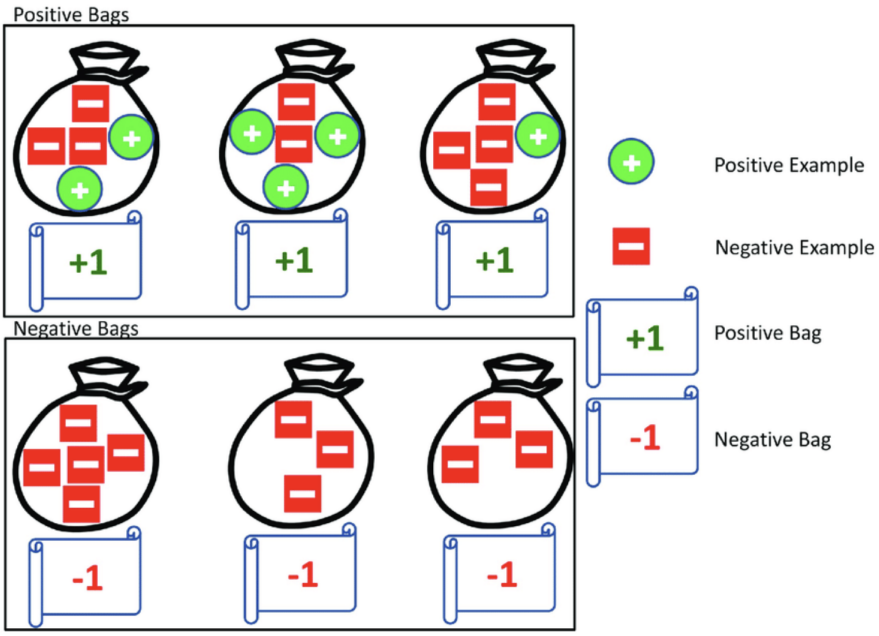

Figure 4

Illustration of the concept of bags and instances (or "examples") in MIL (Source: https://www.researchgate.net/figure/An-illustration-of-the-concept-of-multiple-instance-learning-In-MIL-training-examples_fig1_315925709)

Applied to our specific usage, we predict to treat each sketch as "bags" and each treat each component as "instances." The learner's goal is to produce a concept that will label individual instances (components) correctly in the future.

Responsibility of each team member

Aakash Madabhushi:

- Working on validation and testing data on the Computer Vision model
- Working on the preprocessing and data collection

Prince Hodonou:

- Working on training with weakly labeled sketches
- Working on generating code from network classification