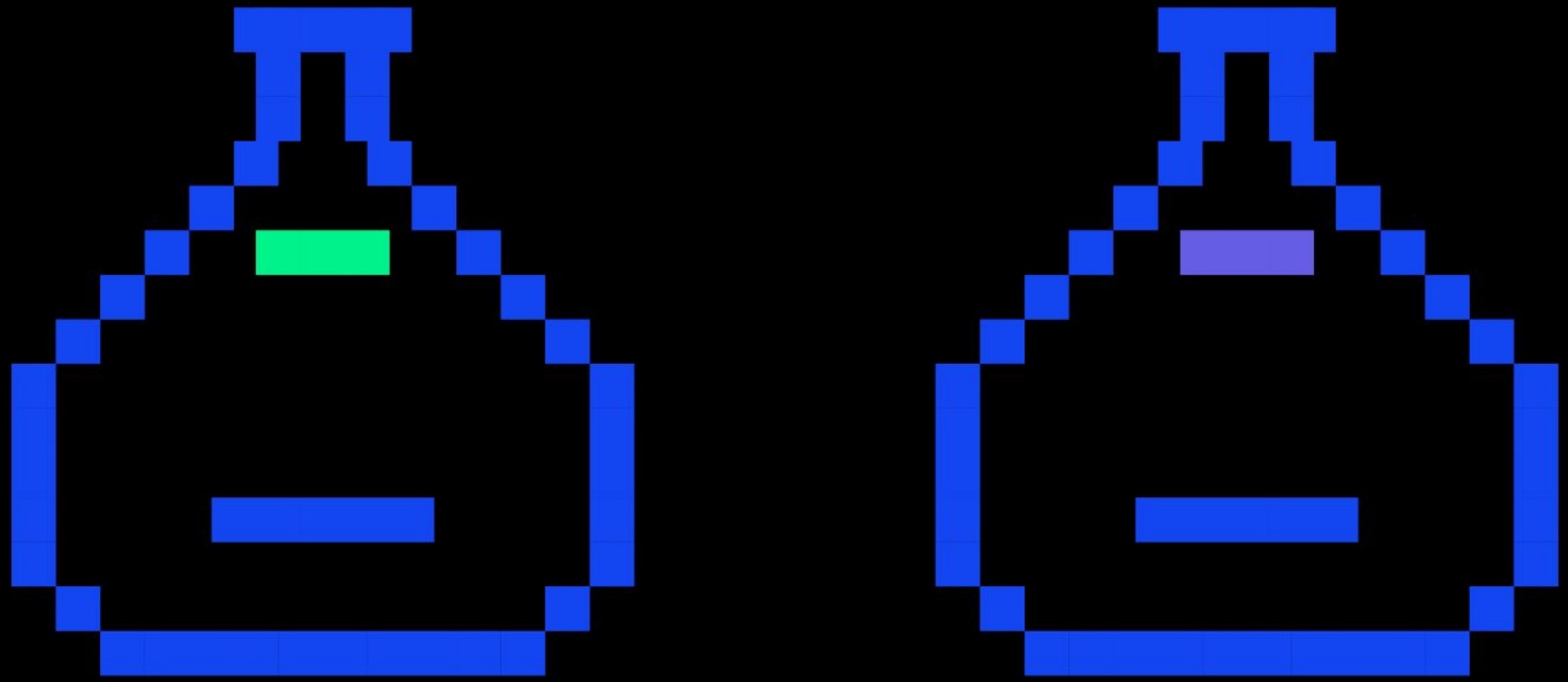
CLOUDFLIGHT
CODING
CONTEST

Level 2



CLOUDFLIGHT
CODING
CONTEST

Level 2

- At some point in your production line you need to separate items based upon their labels. The two classes of items are processed differently later in the production line
- At the moment you have a person doing this by manually inspecting items. However, as you want to scale up your production, your employee cannot handle the workload any more
- To manage the workload, we would like to introduce an AI system, that can automatically differentiate between the two types of items. The worker will then only need to pick out items that have been classified incorrectly and put them to the other production sub-line. We need a minimum overall accuracy of 93% to achieve the necessary speed-up

CLOUDFLIGHT
CODING
CONTEST

Level 2

- You already have in place a computer vision system, that takes a photo of the label on the product and send it to your system
- The dimensions of images that you get and are about to process are 28x84 pixels

0	1	2	3	4	…		81	82	83
84	85	86	87		…			166	167
168	169				…				
…	…	…	…	…	…	…	…	…	…
					…				
					…		2349	2350	2351

CLOUDFLIGHT CODING CONTEST

Level 2

0	1	2	3	4	…		81	82	83
84	85	86	87		…			166	167
168	169				…				
…	…	…	…	…	…	…	…	…	…
					…				
					…		2349	2350	2351

- In your dataset, this array is flattened into a vector of size 2352
- Here are two of the images, that you can also find in your training set. First one has a label 0, while the second one has a label 1

CLOUDFLIGHT
CODING
CONTEST

Level 2

- Input data:
  - Training images of production item labels with one single mulit-digit number per image and the corresponding classes
    - Format:

      N (integer) - number of examples

      pixel_values (integer values separated by commas) (repeated N times, each example in its own line)- includes 2352 integers that correspond to pixel values of a flattened 28x84 image

      class (integer) (repeated N times, each example in its own line) - integer (0 or 1) corresponding to a correct class, given in the same order as the pixel_values
  - Test images (without the desired classes)
    - Format:

      M (integer) - number of examples

      pixel_values (integer values separated by commas) (repeated N times, each example in its own line) - includes 2352 integers that correspond to pixel values of a flattened 28x84 image
- Output
  - Predicted classes of the production items in the test images
    - Format:

      label (integer) (repeated M times, each example in its own line) - integer (0 or 1) corresponding to a predicted class, given in the same order as the pixel_values
- Evaluation
  - 93% of all images must be classified correctly in order to pass the level
  - All miss-classifications have the same cost