Week 4

- This week would be using a webcam to capture images and classifying the images

Building Web Page

- The first step would be building a web page and add the webcam
- Here is the webcam Java Script class:

```
// Copyright 2019 The TensorFlow Authors. All Rights Reserved.
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
//     http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
// =============================================================================

/**
 * A class that wraps webcam video elements to capture Tensor4Ds.
 */


class Webcam {
  /**
   * @param {HTMLVideoElement} webcamElement A HTMLVideoElement representing the
   *     webcam feed.
   */
  constructor(webcamElement) {
    this.webcamElement = webcamElement;
  }

  /**
   * Captures a frame from the webcam and normalizes it between -1 and 1.
   * Returns a batched image (1-element batch) of shape [1, w, h, c].
   */
```

```
  capture() {
    return tf.tidy(() => {
      // Reads the image as a Tensor from the webcam <video> element.
      const webcamImage = tf.browser.fromPixels(this.webcamElement);

      const reversedImage = webcamImage.reverse(1);

      // Crop the image so we're using the center square of the rectangular
      // webcam.
      const croppedImage = this.cropImage(reversedImage);

      // Expand the outer most dimension so we have a batch size of 1.
      const batchedImage = croppedImage.expandDims(0);

      // Normalize the image between -1 and 1. The image comes in between 0-255,
      // so we divide by 127 and subtract 1.
      return batchedImage.toFloat().div(tf.scalar(127)).sub(tf.scalar(1));
    });
  }

  /**
   * Crops an image tensor so we get a square image with no white space.
   * @param {Tensor4D} img An input image Tensor to crop.
   */


  cropImage(img) {
    const size = Math.min(img.shape[0], img.shape[1]);
    const centerHeight = img.shape[0] / 2;
    const beginHeight = centerHeight - (size / 2);
    const centerWidth = img.shape[1] / 2;
    const beginWidth = centerWidth - (size / 2);
    return img.slice([beginHeight, beginWidth, 0], [size, size, 3]);
  }

  /**
   * Adjusts the video size so we can make a centered square crop without
   * including whitespace.
   * @param {number} width The real width of the video element.
   * @param {number} height The real height of the video element.
   */


  adjustVideoSize(width, height) {
    const aspectRatio = width / height;
    if (width >= height) {
      this.webcamElement.width = aspectRatio * this.webcamElement.height;
    } else if (width < height) {
      this.webcamElement.height = this.webcamElement.width / aspectRatio;
    }
  }

  async setup() {
    return new Promise((resolve, reject) => {
      navigator.getUserMedia = navigator.getUserMedia ||
          navigator.webkitGetUserMedia || navigator.mozGetUserMedia ||
          navigator.msGetUserMedia;
      if (navigator.getUserMedia) {
```

```
        navigator.getUserMedia(
            {video: {width: 224, height: 224}},
            stream => {
              this.webcamElement.srcObject = stream;
              this.webcamElement.addEventListener('loadeddata', async () => {
                this.adjustVideoSize(
                    this.webcamElement.videoWidth,
                    this.webcamElement.videoHeight);
                resolve();
              }, false);
            },
            error => {
              reject(error);
            });
      } else {
        reject();
      }
    });
  }
}
```

- Here is the simple web page created to capture the content of the webcam

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e41c5d43-bc63-4cde-b198-b3b9bfcfaaa1/SimpleWebPage.zip

Creating Rock Paper Scissors Classifier

- This is the code for the Rock paper scissors classifier

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ceb38071-408c-46ae-9f11-28c425547abd/CaptureData.zip

Exercise

- Code for exercise

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5f11ed6d-a1b2-4e9f-bc36-1202777c2a08/Exercise.zip