Model documentation and write-up

1. Who are you (mini-bio) and what do you do professionally?

   AZK90: I am a PhD student at ___. My area of focus is Disaster Resilience which is a part of Civil Engineering department. My work involves employing probabilistic methods and machine learning approaches for assessment of natural hazards. For example, I use different machine learning methods such as Artificial Neural Networks for prediction of storm surge and down-scaling of future climate projections for precipitation frequency analysis.

   kaveh9877: I am a PhD student at ___ and I work in area of Disaster Resilience. My main research is focused on risk assessment of hazards such as flood and storm surge. I am also working on employing geospatial analysis and machine learning approaches in my research.

2. What motivated you to compete in this challenge?

Due to our research, we became acquainted with machine learning methods such as Artificial Neural Networks and Gaussian Process Regression, etc. We were interested to learn more about these methods so as a next step we started to study deep learning recently. We thought this skill might be useful in our research, too. We decided to participate in deep learning challenges because we believed that it can help us to learn these methods more effectively.

3. High level summary of your approach: what did you do and why?

At first, we tried approaches that are mostly used in human action recognition. However, after a while we tried an approach that in our opinion was simple, faster and more suitable to the nature of the videos for recognizing stalled and flowing vessels. The approach that gave us the highest score was converting the videos to combination of images and using transfer learning. The steps of the approach are as follow:

1. At first step, we tried to select a suitable set of input data from the training set which we thought might give us good results.
2. For second step, we split videos to 4 equal groups of frames.
3. In third step, we obtained the average of the pixels of frames in each group and used that as the representative of these frame in our training approach.
4. Then we constructed a single image that consisted of mean of these groups of frames.
5. Afterward, in the last step, we used Deep neural network (with retraining ResNet18) to classify testing videos as stalled or flowing.

4. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

   Maybe the most impactful part of our code is the function that we used for converting the video frames to a single image which consist of four tiles that each represent the average of pixels of ¼ of the video frames. Other than that, other parts of the code are what is usually done in training deep learning models with MATLAB.

   The explanation about this part of the code is as follows:

We used the benchmark code that was provided by MathWorks. The difference is that we divided the video's frames to 4 groups with the same number of frames. Then we found the mean of pixels in each of those groups and converted each of these groups to a single image. Afterwards, we used the imtile function and combined those four images into a single image. The output of this function (which called “video2img”) is actually a 224X224X3 color image.

```
function video2img = readVideo(filename)


vr = VideoReader(filename);
i = 0;


while hasFrame(vr)
      i = i+1;
      framein = readFrame(vr);
      if i < 2
            [Bbox, polymask] = detectROI(framein);
      end
      frame = bsxfun(@times, framein, cast(polymask,class(framein)));
      frame = imcrop(frame, Bbox);
      frame=imresize(frame,[224,224]);
      video(:,:,:,i)=frame;
end
num=floor(i/4);
for ii=1:4
      x=1+(ii-1)*num;
      y=ii*num;
      vidsection=video(:,:,:,x:y);
      vids{ii}=imresize(uint8(mean(vidsection,4)),[112 112]);
end
video2img=imtile({vids{1},vids{2},vids{3},vids{4}});
video2img=rgb2gray(video2img);
video2img= ind2rgb(video2img, hsv);
%imshow(video2img)
end
```

5. Please provide the machine specs and time you used to run your model.
   - CPU (model): Intel Core i7
   - GPU (model or N/A): Nvidia GTX960 4GB Memory
   - Memory (GB): 16GB
   - OS: Windows 10
   - Train duration: It should take less than 5 hours

- Inference duration: About 5 minutes

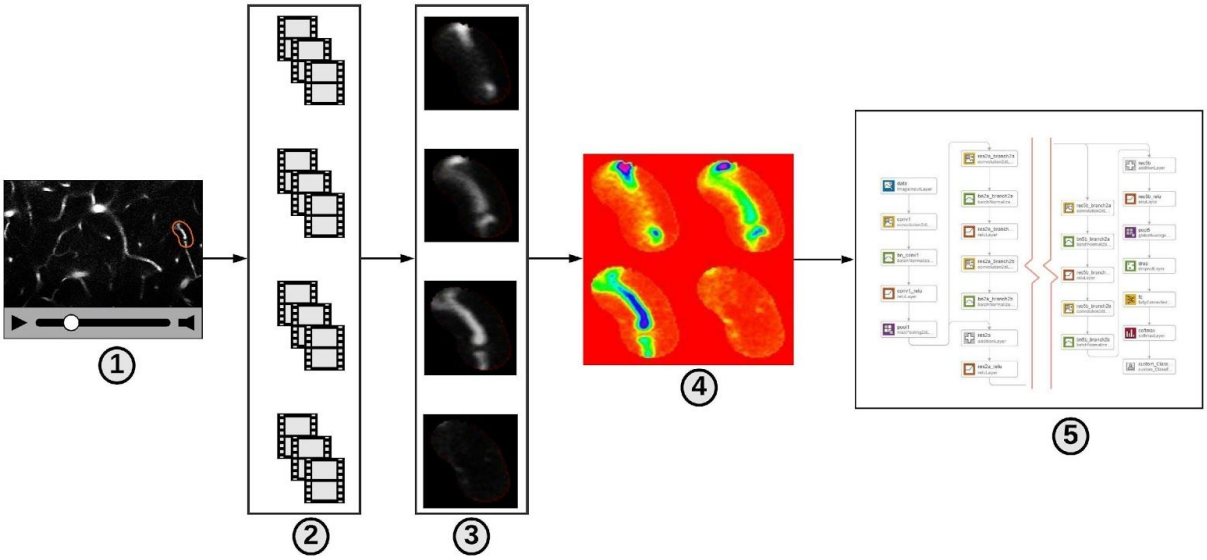
6. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?
   - First, we used the benchmark code approach which involved extracting feature with pre-trained network and then using Long Short-Term Memory (LSTM) network for predicting the labels. We used different options and different pretrained deep neural network and we also developed a bigger training set than Nano and Micro. These changes improved the score. After that, we used opticalFlow object to obtain pixel velocity magnitudes image. We used these along with previous data as an input for LSTM network, however, we could not improve the results. We also explored other ways of combining Convolutional Neural Network (CNN) and LSTM network (other than the benchmark code method), but we were not able to improve the score.
   - The second approach that we used was converting videos to 3D images and using 3D CNN for classifying. We tried simple networks with few numbers of convolution3dLayer, but the score was not improved in the way that we used 3D CNN.

7. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?
   No, we only used MATLAB toolboxes.

8. How did you evaluate performance of the model other than the provided metric, if at all?
   We only used Matthew's correlation coefficient.

9. Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?
   Sometimes, the readVideo function output does not capture the ROI correctly (or it captures it differently). Mostly this happens when we run this code on different computers, however, it also rarely happened when we used the same computer (See section 2.8.2 of "main_file.mlx")

10. Do you have any useful charts, graphs, or visualizations from the process?
    Each number in the following figure is explained in question 3 as steps of our approach.



11. If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?
    The highest score that we obtained is based on retraining ResNet18 on just 13990 selected video files from the main training dataset which consisted of 573050 videos. If we were to continue working on this project, we would have used other pre-trained networks and also, we would include other parts of dataset. Also, we probably should consider combining our highest score method with pre-trained 3D CNN.