SharpCam

Group D1: Nathan Koch, Sean Pogorelc and Rebecca Jean-Louis

Use Case

- Nearly everyone has a smartphone these days, which allows anyone to record almost any scenario, unfortunately this video usually has some blur from the users handshake

- SharpCam allows a user to record a video and then have it post-processed to remove any blur from the original recording

- This allows the cameraman to record a clean stable video in high action events like sports, or have an easy software replacement to physical devices that reduce camera motion such as a gimbal

ECE Areas:

- **Signals and Systems**: Frame Deblurring Process, Multiple Frame Alignment, Optimization Solutions
- **Software/Embedded**: Designing a CNN, Multithreading A CNN Model

Requirements

- Deblur a single 720p frame within 3 seconds
  - **Challenges:**
    - Creating a CNN that can take in 5 frames @ 720p and learn the alignment of all of the images while also learning to deblur the given frame based on the prior 4 frames.
    - Gathering training data for our CNN
    - Multithreading our multiple Jetson boards so that we can have multiple sets of frames being de-blurred by our CNN at the same time
- Upload the deblurred video immediately after post-processing to the users SD card
  - **Challenges:**
    - Re-constructing the video with the new deblurred frames and the original sharp frames
    - Compressing our raw image data so that it can reasonably fit on the users SD card.

Requirements

- Record and deblur an additional video immediately after the first video has been processed
  - **Challenges:**
    - Ensuring that the two Jetson boards can properly reset and remove all erroneous information when the user has already recorded a video and wishes to record another
    - Use the Jetson board GPIO pins to send a signal to multiple different LED lights to indicate whether the video is currently recording, post processing, or that the camera is on
- Have a validation error of no more 30%
  - **Challenges:**
    - Gathering training data for our CNN
    - Developing a process for measuring the sharpness of an image that is time efficient

Requirements

- Record video at 720p, 30fps
  - **Challenges:**
    - Downsampling our 60fps camera s.t. we get a clean 30fps video that doesn’t have obvious jumps or skips. Want the video to look like it was filmed at 30fps natively

- LED indicators to convey information about the state of the camera to the user
  - **Challenges:**
    - Same as a challenge on the previous slide, using the Jetson board GPIO pins to send a signal to multiple different LED lights to indicate whether the video is currently recording, doing post processing, or that the camera is on

Requirements

- The entire system is able to be handheld and operable inside a fabricated frame
  - **Challenges**:
    - 3D modeling and printing a container that comfortably fits all of the camera, board, and power elements
- Contains a SD card capable of storing at least 5 minutes of total video
  - **Challenges:**
    - Compressing the raw video s.t. it takes up a reasonable amount of space on the SD card
    - Storing video to the SD card from the Jetson Nano

Solution Approach (Hardware)

- Jetson Nano
  - Using 2 of these to handle our CNN processing due to its 128-core Maxwell GPUs
- e-CAM30_CUNANO
  - Can film at 1280x720p @ 60fps, we will downsample to 30fps though for computational reasons
- Breadboard and LEDs
  - Used in conjunction with the Jetson Nanos as our UI to tell the user if the camera is on, the system is recording, and/or the system is currently processing video

Solution Approach (Software)

- Matlab
  - Used to perform our Video Processing as its the most efficient that we know of to perform this task
- Python
  - Used for our backend development, we plan on leveraging OpenCV to assist with our handling of pre/post processed frames.
- We plan on using 1 board as the main source of communication with other parts of the system (Camera, SD card, LEDs) while the other will be asynchronously connected and used solely for Video processing purposes

Testing, Verification, and Metrics

- Training Set Data
    - Composed of 2 sets of frames, blurred and sharp
        - Will be a combination of publicly available images and our own that we create by synthetically blurring
        - Frames in each set will have a corresponding image in the other set
    - Having both allows us to check how well our deblurring algorithm works, so we will make this our standard for performance
        - Obvious goal is for the deblurring algorithm to get the blurred images as close as possible to their sharp counterparts

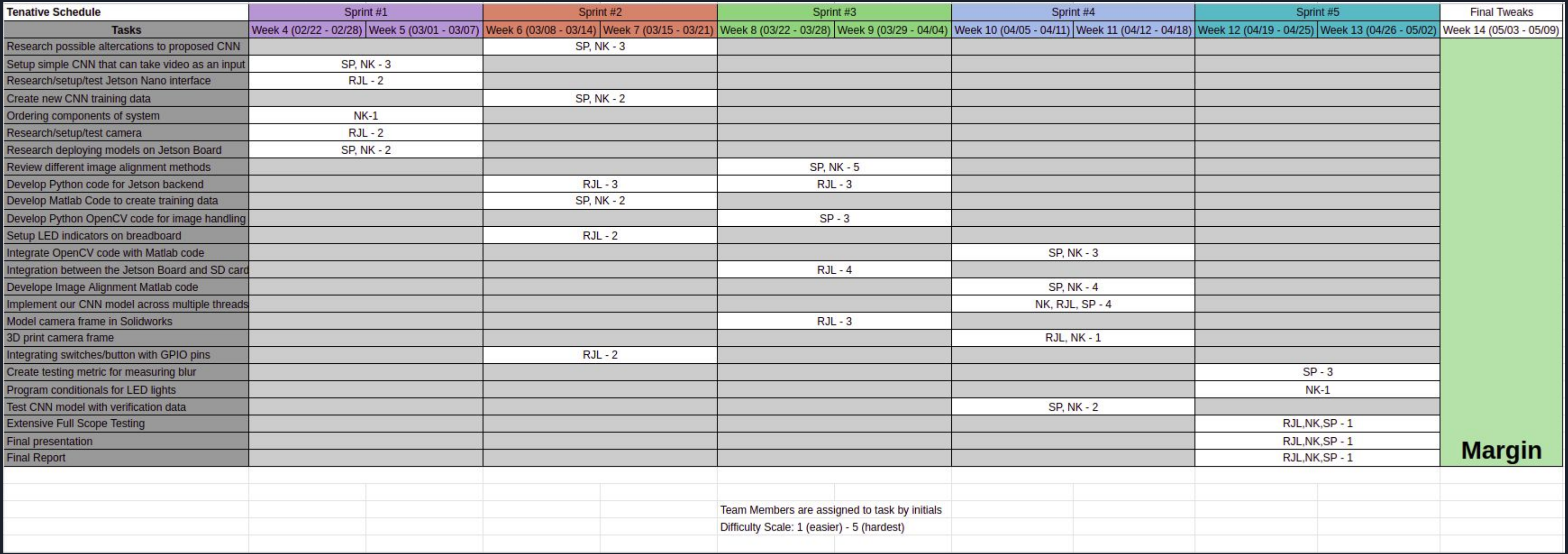
Testing, Verification, and Metrics

- Deblurring Metric
  - Use a HPF to assign ‘smoothness’ ratings
  - Use these ratings to either indicate that we detect a noticeable improvement post-processing or not
- SD Card and Video Quality
  - Take video using camera hooked up to laptop
  - Take video (>5 minutes) with our system
  - Offload video from SD card and compare for any difference in picture quality/check to make sure video is right length

Task/Division of Labor

Tasks	Sean	Nathan	Rebecca
Create new training Data for CNN	✓	✓	
Integration between the Jetson Boards and Camera	✓		✓
Program alignment process for multiple frames		✓	
Deploying python/MATLAB code on Jetson Boards	✓		✓
Developing/deploying CNN model	✓	✓	
Create a 3D model for to encapsulate the camera/board for handheld use		✓	✓
Integration between the Jetson Board and SD card			✓
Developing deblurring metric and quantifying it	✓	✓	

Schedule



Tenative Schedule	Sprint #1		Sprint #2		Sprint #3		Sprint #4		Sprint #5		Final Tweaks
Tasks	Week 4 (02/22 - 02/28)	Week 5 (03/01 - 03/07)	Week 6 (03/08 - 03/14)	Week 7 (03/15 - 03/21)	Week 8 (03/22 - 03/28)	Week 9 (03/29 - 04/04)	Week 10 (04/05 - 04/11)	Week 11 (04/12 - 04/18)	Week 12 (04/19 - 04/25)	Week 13 (04/26 - 05/02)	Week 14 (05/03 - 05/09)
Research possible altercations to proposed CNN			SP, NK - 3								
Setup simple CNN that can take video as an input	SP, NK - 3										**Margin**
Research/setup/test Jetson Nano interface	RJL - 2										
Create new CNN training data			SP, NK - 2								
Ordering components of system	NK-1										
Research/setup/test camera	RJL - 2										
Research deploying models on Jetson Board	SP, NK - 2										
Review different image alignment methods					SP, NK - 5						
Develop Python code for Jetson backend			RJL - 3		RJL - 3						
Develop Matlab Code to create training data			SP, NK - 2								
Develop Python OpenCV code for image handling					SP - 3						
Setup LED indicators on breadboard			RJL - 2								
Integrate OpenCV code with Matlab code							SP, NK - 3				
Integration between the Jetson Board and SD card					RJL - 4						
Develope Image Alignment Matlab code							SP, NK - 4				
Implement our CNN model across multiple threads							NK, RJL, SP - 4				
Model camera frame in Solidworks					RJL - 3						
3D print camera frame							RJL, NK - 1				
Integrating switches/button with GPIO pins			RJL - 2								
Create testing metric for measuring blur									SP - 3		
Program conditionals for LED lights									NK-1		
Test CNN model with verification data							SP, NK - 2				
Extensive Full Scope Testing									RJL,NK,SP - 1		
Final presentation									RJL,NK,SP - 1		
Final Report									RJL,NK,SP - 1		

Team Members are assigned to task by initials

Difficulty Scale: 1 (easier) - 5 (hardest)