POKERCAM

Spring 2021 - Team E7
Jeremy Klotz, Ethan Rich, and Sid Domala

Application Area

Most professional card games analyses are not automated

Our system

- Images cards as they are dealt
- Provides a web interface to visualize hands
- Does not require card deck preparation

Application

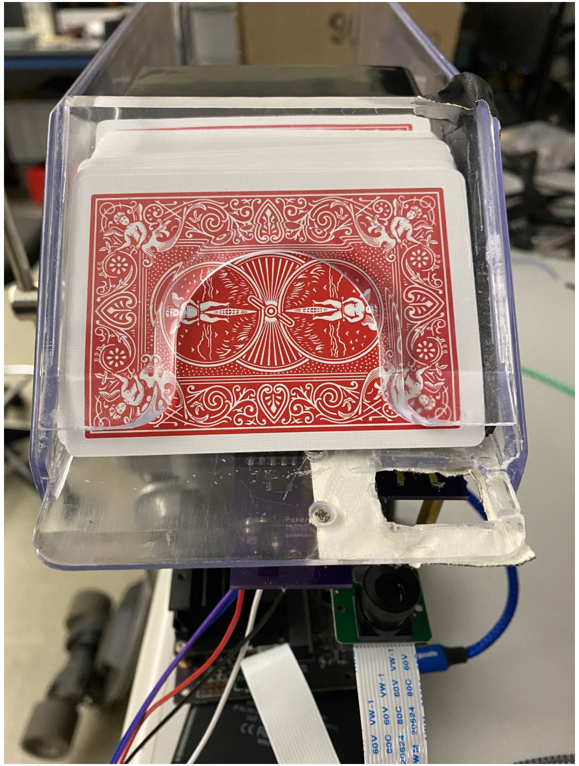
- Commentators/Analysts
- TV spectators

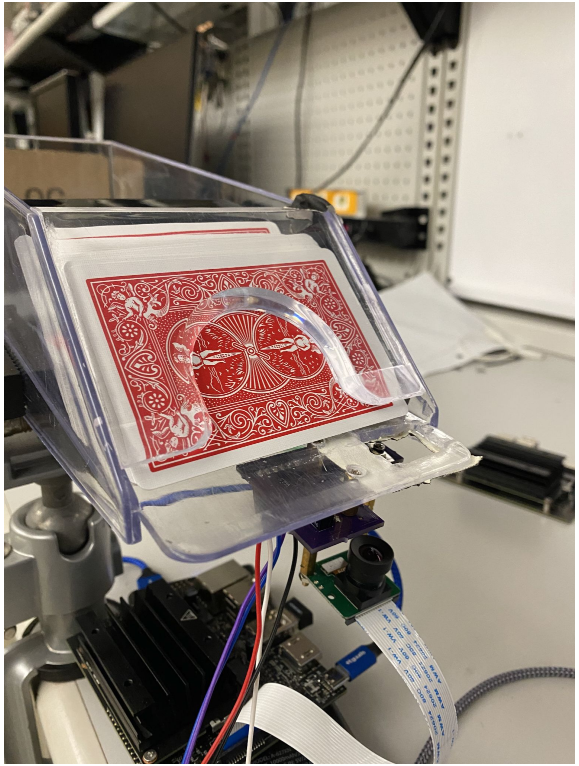
Complete Solution

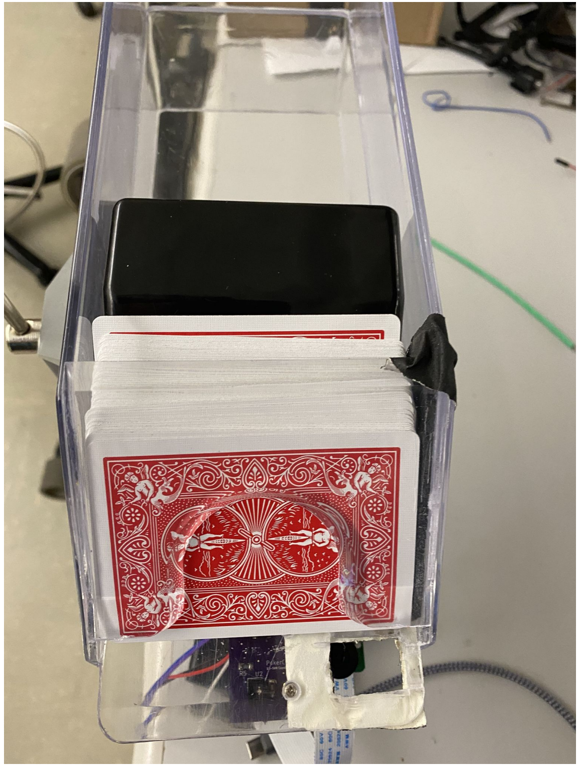
Final demo will simulate multiple rounds of a card game

- One individual will withdraw and deal cards from the card shoe
- Computer visualizes captured image and timing statistics
- Web app will showcase classification result and allow user input to change game state

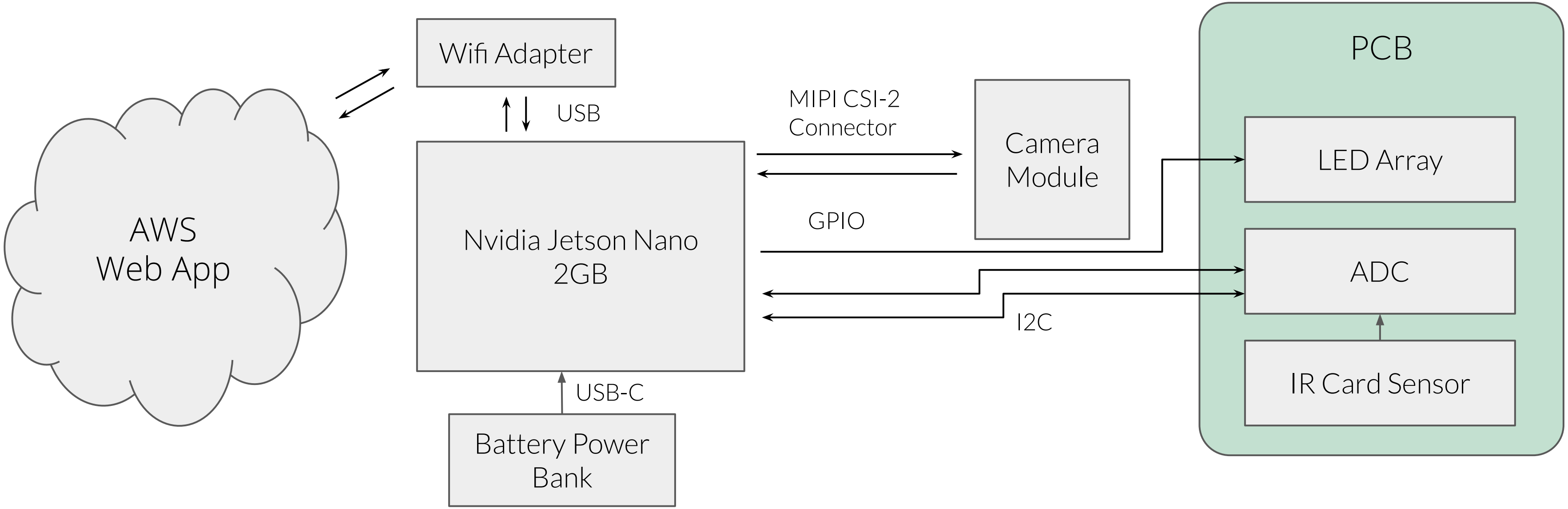
Cardshoe





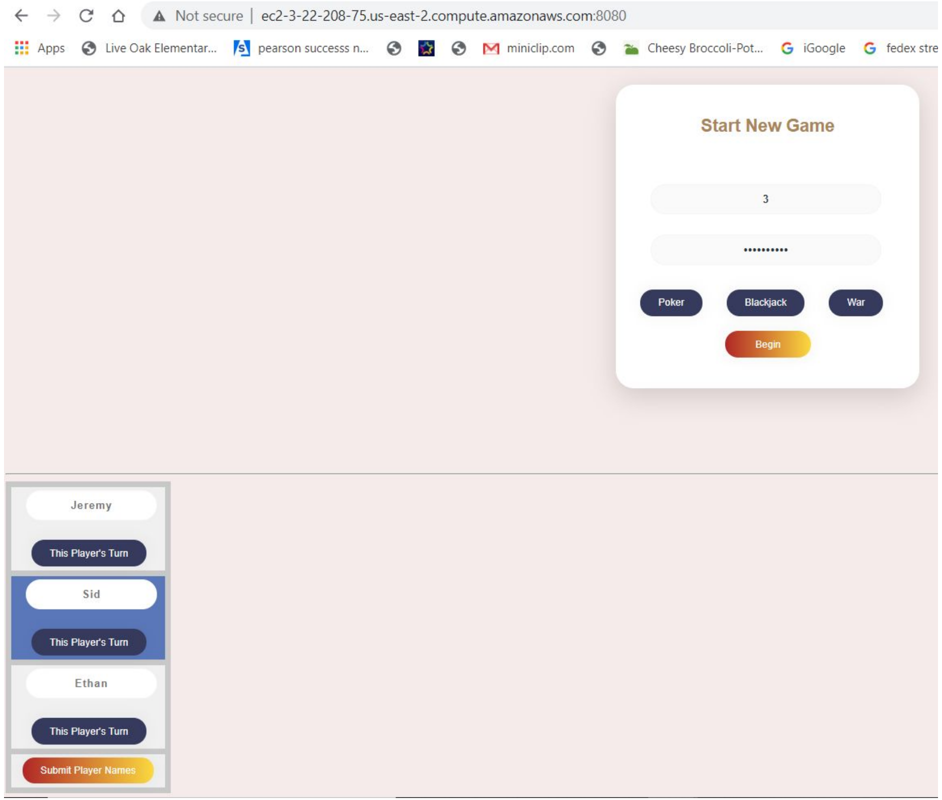


System Specification

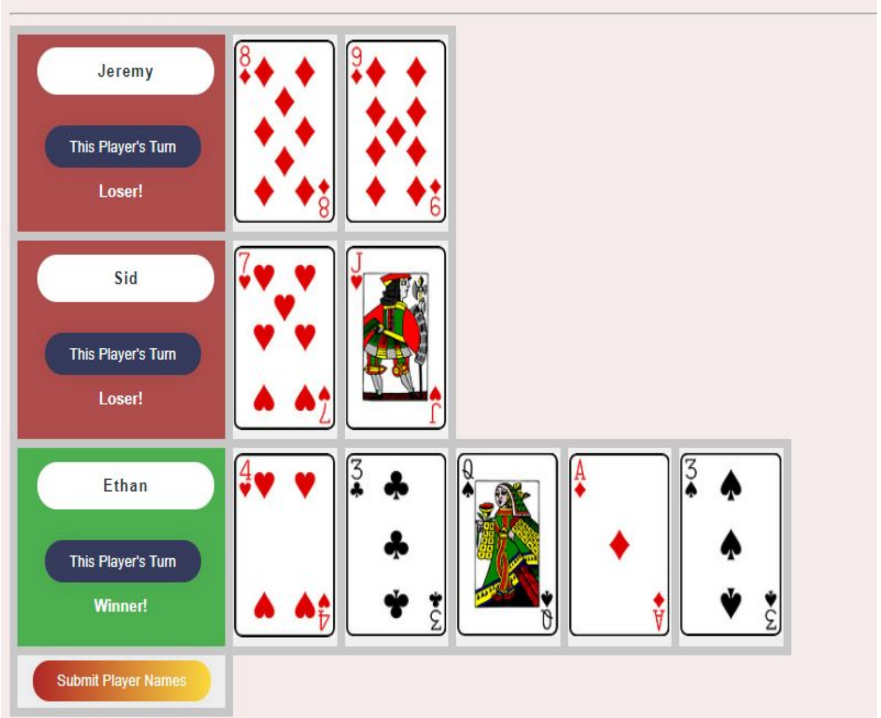


Web App - Blackjack

Start of Game



End of Game

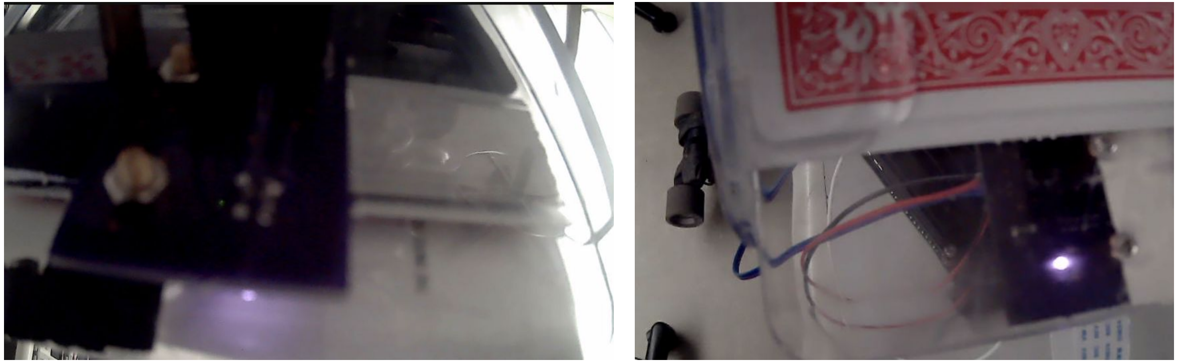


Design Requirements

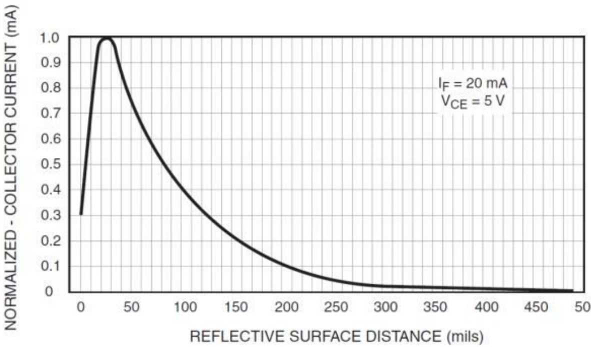
Requirements	Desired Results	Actual Results
Classification Accuracy	≥ 98%	Test Accuracy: **98.1%**
Classification Latency	≤ 2s for web app to update	Average: **0.18s** Maximum: **0.22s**
Battery Life	≥ 2 hours	**In testing**
Sufficient Memory	Classify entire card deck in ≤ 104 seconds	Classified entire card deck in **54 seconds**
False Triggers	0 False Triggers	**In testing**

Design Tradeoffs

- Image colorspace
- IR Reflectance vs. ‘Beam-Break’ Trigger
- Web Application Framework
  - Flask vs. Django
- Database
  - MongoDB vs SQL

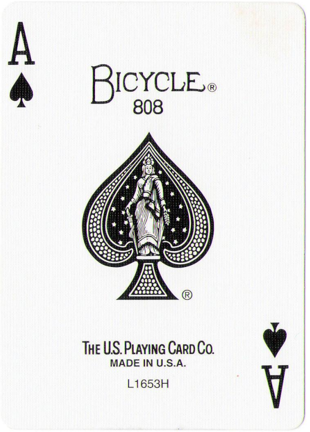


Infrared reflectance shown in webcam without IR filter

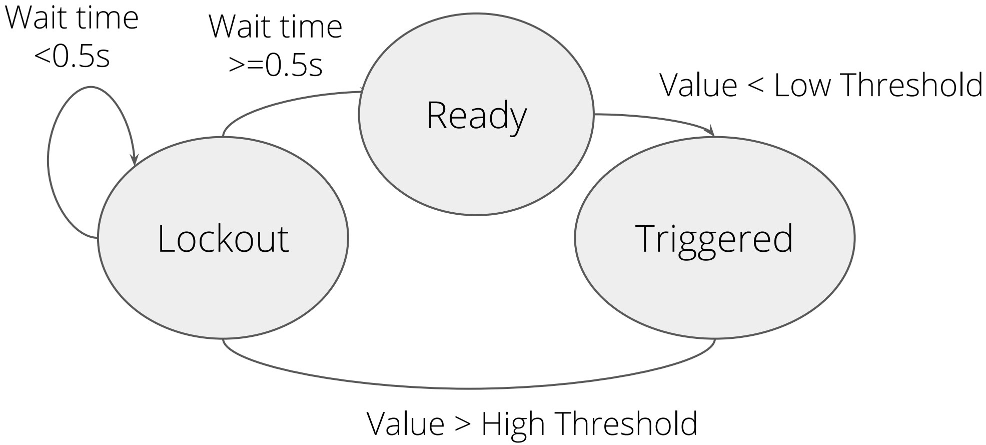


QRD1114 Distance vs. I_C

Design Tradeoff: Trigger Design



Failure case for initial trigger implementation



Updated finite state machine to eliminate double triggers

Testing Strategy

- **Subsystem Testing**
  - **Image Classification:** Image 4 Bicycle Standard card decks over multiple days (split into training, validation, and testing datasets)
  - **Hardware:** Manually draw cards from shoe and note any false/failed triggers
  - **Web Application:** Measure latency between RESTful API request and web app update
- **Full System Testing**
  - Deal new, unopened Bicycle card deck for at least 2 hours
  - Record ground-truth labels of cards, watch web display for false triggers, and measure latency between trigger and web display update

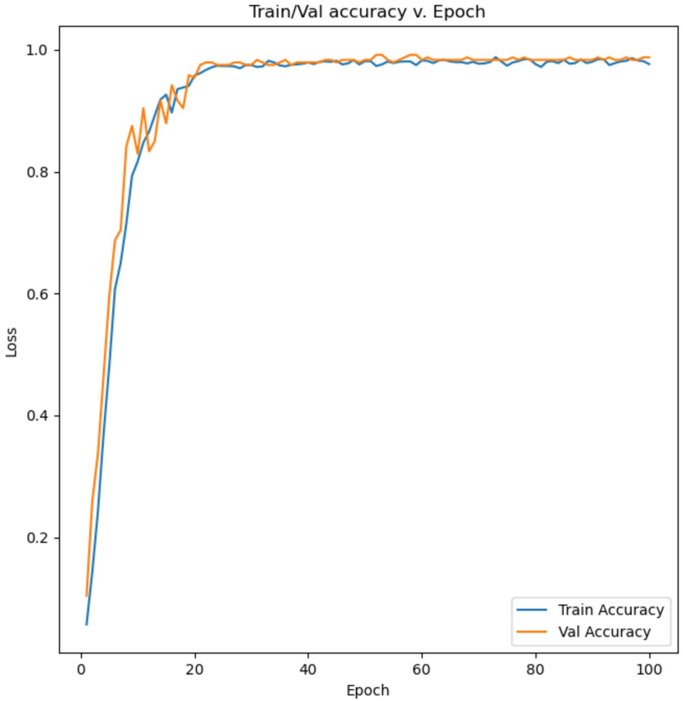
Image Classification

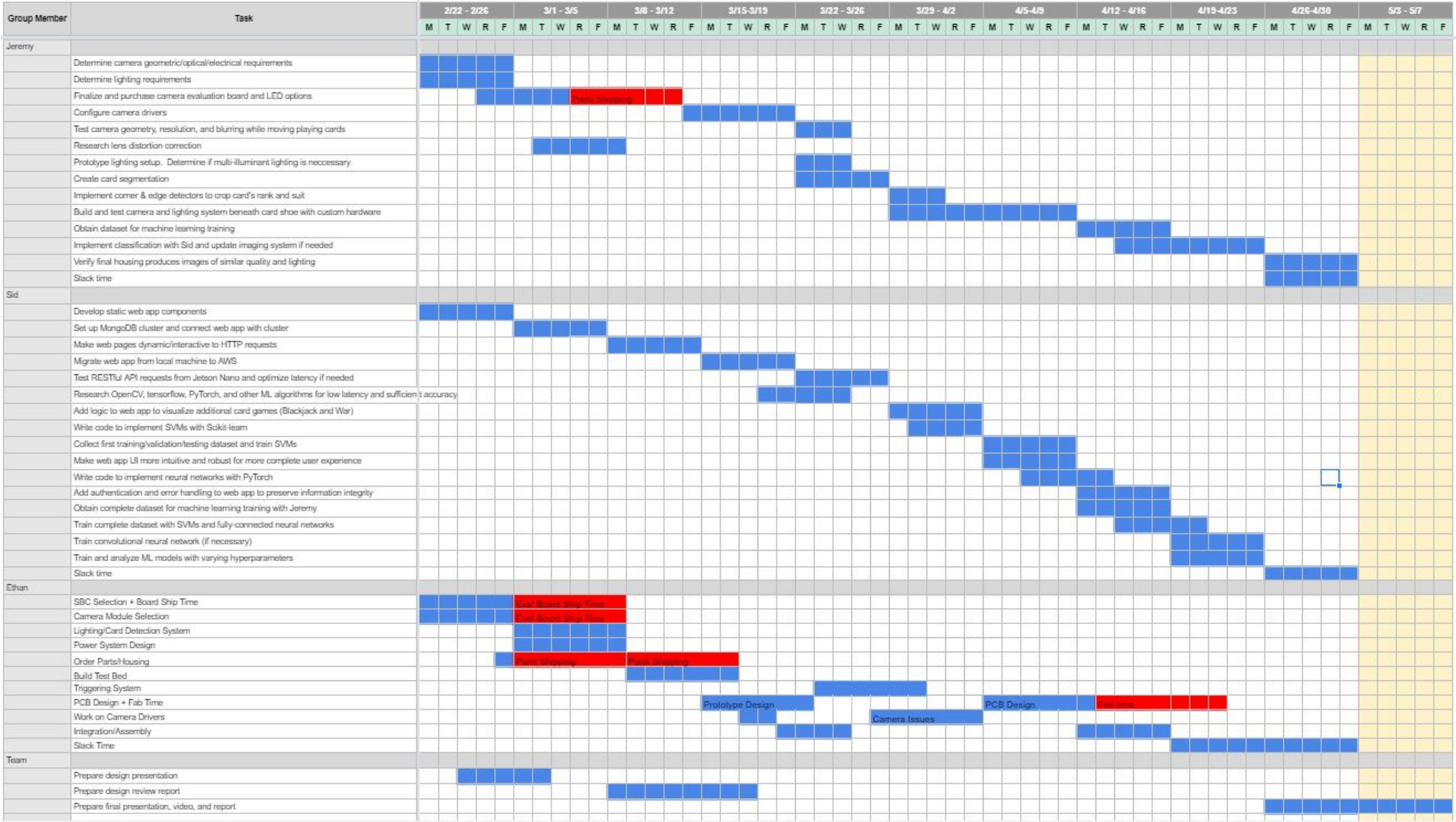
- Model selection
- Single network predicts rank and suit probabilities
- | training set | = 1906

  | validation set | = 272

  | testing set | = 546
- Training set includes random crops
- **98.1% test accuracy**

Example classifier input
(ground truth: 6 of clubs)





Group Member	Task	2/22 - 2/26					3/1 - 3/5					3/8 - 3/12					3/15-3/19					3/22 - 3/26					3/29 - 4/2					4/5-4/9					4/12 - 4/16					4/19-4/23					4/26-4/30					5/3 - 5/7				
		M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
Jeremy																																																								
	Determine camera geometric/optical/electrical requirements	🟦	🟦	🟦	🟦	🟦																																																		
	Determine lighting requirements	🟦	🟦	🟦	🟦	🟦																																																		
	Finalize and purchase camera evaluation board and LED options				🟦	🟦	🟦	🟦	🟦	Parts Shipping	🟥	🟥	🟥	🟥	🟥																																										
	Configure camera drivers															🟦	🟦	🟦	🟦	🟦	🟦																																				
	Test camera geometry, resolution, and blurring while moving playing cards																					🟦	🟦	🟦																																
	Research lens distortion correction							🟦	🟦	🟦	🟦	🟦																																												
	Prototype lighting setup. Determine if multi-illuminant lighting is neccessary																					🟦	🟦	🟦																																
	Create card segmentation																					🟦	🟦	🟦	🟦	🟦																														
	Implement corner & edge detectors to crop card's rank and suit																										🟦	🟦	🟦																												
	Build and test camera and lighting system beneath card shoe with custom hardware																										🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																				
	Obtain dataset for machine learning training																																				🟦	🟦	🟦	🟦	🟦														
	Implement classification with Sid and update imaging system if needed																																						🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦										
	Verify final housing produces images of similar quality and lighting																																														🟦	🟦	🟦	🟦	🟦					
	Slack time																																														🟦	🟦	🟦	🟦	🟦					
Sid																																																								
	Develop static web app components	🟦	🟦	🟦	🟦	🟦																																																		
	Set up MongoDB cluster and connect web app with cluster						🟦	🟦	🟦	🟦	🟦																																													
	Make web pages dynamic/interactive to HTTP requests											🟦	🟦	🟦	🟦	🟦																																								
	Migrate web app from local machine to AWS																🟦	🟦	🟦	🟦	🟦																																			
	Test RESTful API requests from Jetson Nano and optimize latency if needed																					🟦	🟦	🟦	🟦	🟦																														
	Research OpenCV, tensorflow, PyTorch, and other ML algorithms for low latency and sufficient accuracy																			🟦	🟦	🟦	🟦	🟦																																
	Add logic to web app to visualize additional card games (Blackjack and War)																										🟦	🟦	🟦	🟦	🟦																									
	Write code to implement SVMs with Scikit-learn																											🟦	🟦	🟦	🟦																									
	Collect first training/validation/testing dataset and train SVMs																															🟦	🟦	🟦	🟦	🟦																				
	Make web app UI more intuitive and robust for more complete user experience																															🟦	🟦	🟦	🟦	🟦																				
	Write code to implement neural networks with PyTorch																																	🟦	🟦	🟦	🟦	🟦																		
	Add authentication and error handling to web app to preserve information integrity																																				🟦	🟦	🟦	🟦	🟦															
	Obtain complete dataset for machine learning training with Jeremy																																				🟦	🟦	🟦	🟦	🟦															
	Train complete dataset with SVMs and fully-connected neural networks																																						🟦	🟦	🟦	🟦	🟦													
	Train convolutional neural network (if necessary)																																									🟦	🟦	🟦	🟦	🟦										
	Train and analyze ML models with varying hyperparameters																																									🟦	🟦	🟦	🟦	🟦										
	Slack time																																														🟦	🟦	🟦	🟦	🟦					
Ethan																																																								
	SBC Selection + Board Ship Time	🟦	🟦	🟦	🟦	🟦	Eval Board Ship Time	🟥	🟥	🟥	🟥	🟥																																												
	Camera Module Selection	🟦	🟦	🟦	🟦	🟦	Eval Board Ship Time	🟥	🟥	🟥	🟥	🟥																																												
	Lighting/Card Detection System						🟦	🟦	🟦	🟦	🟦	🟦																																												
	Power System Design						🟦	🟦	🟦	🟦	🟦	🟦																																												
	Order Parts/Housing					🟦	Parts Shipping	🟥	🟥	🟥	🟥	🟥	Parts Shipping	🟥	🟥	🟥	🟥	🟥																																						
	Build Test Bed												🟦	🟦	🟦	🟦	🟦	🟦																																						
	Triggering System																						🟦	🟦	🟦	🟦	🟦	🟦																												
	PCB Design + Fab Time																Prototype Design	🟦	🟦	🟦	🟦	🟦																PCB Design	🟦	🟦	🟦	🟦	🟦	Fab time	🟥	🟥	🟥	🟥	🟥	🟥												
	Work on Camera Drivers																		🟦	🟦						Camera Issues	🟦	🟦	🟦	🟦	🟦																									
	Integration/Assembly																				🟦	🟦	🟦	🟦													🟦	🟦	🟦	🟦	🟦															
	Slack Time																																									🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦					
Team																																																								
	Prepare design presentation			🟦	🟦	🟦	🟦	🟦																																															
	Prepare design review report											🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																																					
	Prepare final presentation, video, and report																																														🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦