Evaluating Image Classification Models for FPGA Board Status Detection

Misha Baitemirova

Stanford University

Background

- LabsLand is an educational technology startup that provides remote access to physical laboratories, including to field-programmable gate array, or FPGA, boards
- A persisting problem has been identifying when LED light indicators on FPGA boards are on and when they are off due to varying light conditions and due to different positions of those boards, that together constitute a challenging condition
- Therefore, our aim for this project was to employ deep learning for computer vision to automate board light status detection of FPGA boards

Problem Statement

- Main goal: classify boards into 11 classes, where 10 classes represent 10 LED indicators that are on, and 1 class represents a board with all LED indicators off
- Use deep learning models for image classification: VGG-16 and EfficientNetB4

Dataset

- Data was generated by capturing photos from a video stream at three different locations at different times of the day to capture varied light conditions
- Some dataset statistics are summarized below:

Location	Number of boards	Total number of images	Training	Validation	Test
Pamplona, Spain	9	3960	2376	792	792
Pamplona, Spain	6	2640	1584	528	528
Seattle, USA	7	3080	1848	616	616

- Examples of boards with an LED indicator on are illustrated below:

Methods

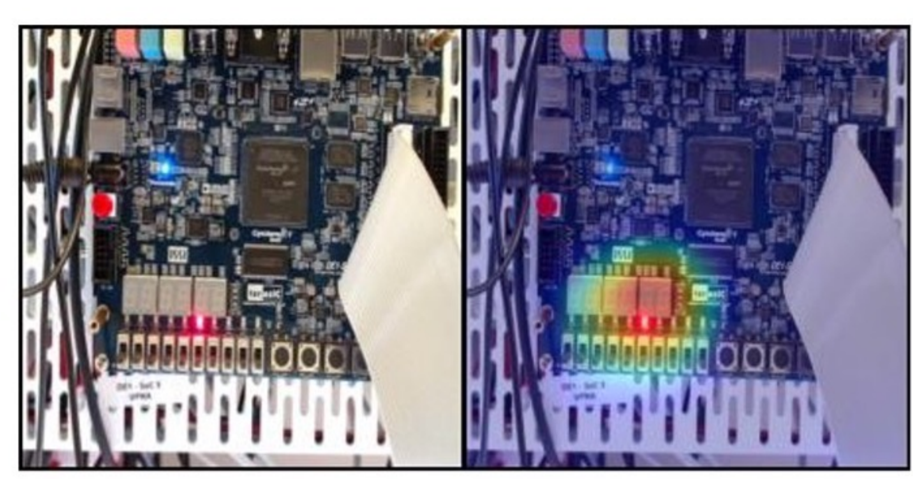
- We used pre-trained VGG-16 and EfficientNetB4 and fine-tuned some layers in a multi-classification task on 11 classes
- We used a learning rate of 1e-5, cross-entropy loss function, batch size of 40, and an Adam optimizer, which is a replacement optimization algorithm for stochastic gradient descent for VGG-16
- For EfficientNetB4, we used default settings as well: learning rate of 1e-5 and an Adam optimizer
- We used Grad-CAM for coarse localization map of the regions deemed important by both models
- We used several affine data transformation techniques, such as random rotation, random resize, and center crop

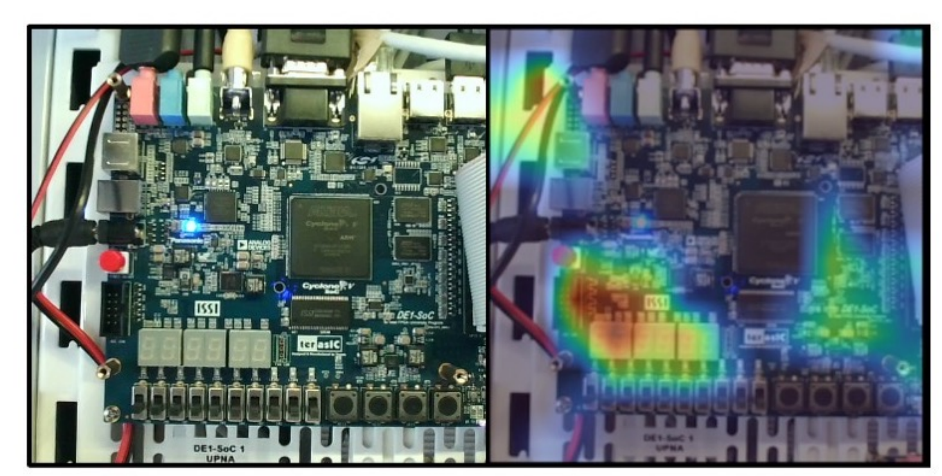
Results

- Accuracy results for both models, with and without data augmentation, are illustrated below:

Augmentation method	VGG-16	EfficientNetB7
None	93.43%	39.97%
Center crop	85.79%	24.81%
Random resize	89.75%	50.76%
Random rotation	94.29%	34.89%
Random color change	93.52%	26.81%
All	88.81%	34.89%

- Grad-CAM visualizations are illustrated below, with VGG-16 Grad-CAM on the left, and EfficientNetB4 Grad-CAM on the right





Conclusions

- Advantages of larger models include larger learning potential, but they are also more prone to over-fitting
- There are advantages to using smaller models, such as EfficientNetB4, that require less computational resources and less training time
- In practice, EfficientNetB4 performed poorly compared to a larger model, VGG-16
- The reason for the poor performance could be the nature of EfficientNet family models

Future Directions

- Testing more sophisticated data augmentation methods, for example deep learning data augmentation methods as not all affine data augmentation were successful
- Hyperparameter tuning to improve the performance of EfficientNetB4 as EfficientNet family models are sensitive to hyperparameter changes
- Using more sophisticated initialization instead of random initialization
- Testing other small image classification models