Fourier Domain Pruning of MobileNet-V2 with Application to Video Based Wildfire Detection

Hongyi Pan, Diaa Badawi, Ahmet Enis Cetin

Electrical and Computer Engineering

University of Illinois at Chicago

Outline

- MobileNet-V2 for Wildfire Detection
- Fourier Transform Based Neural Network Pruning
- Block-Based Image Frame Analysis
- Experimental Results: Neural Network Performance
- Conclusion

MobileNet-V2 for Wildfire Detection

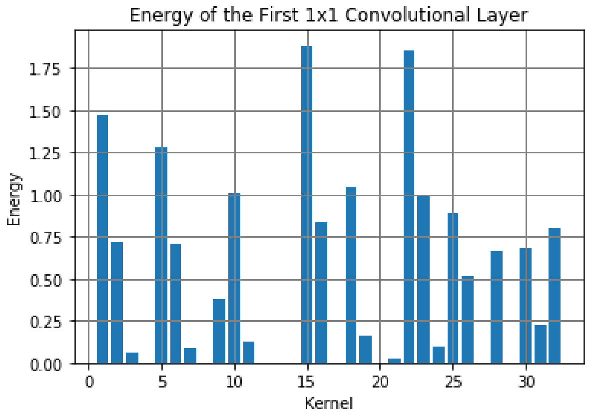
- MobileNet-V2: A mobile CNN model published in 2018.
- We totally have about 10K training images:
  - 3K wildfire images.
  - All images are resized into 224x224.
- We will use NVIDIA Jetson Nano in our wildfire detection system.

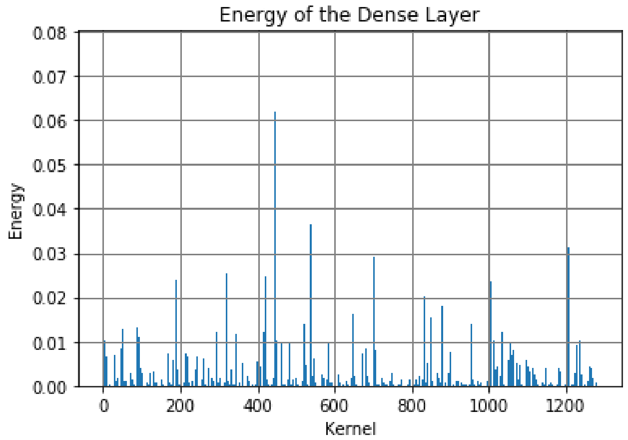
(a) (b) (c) (d) (e)

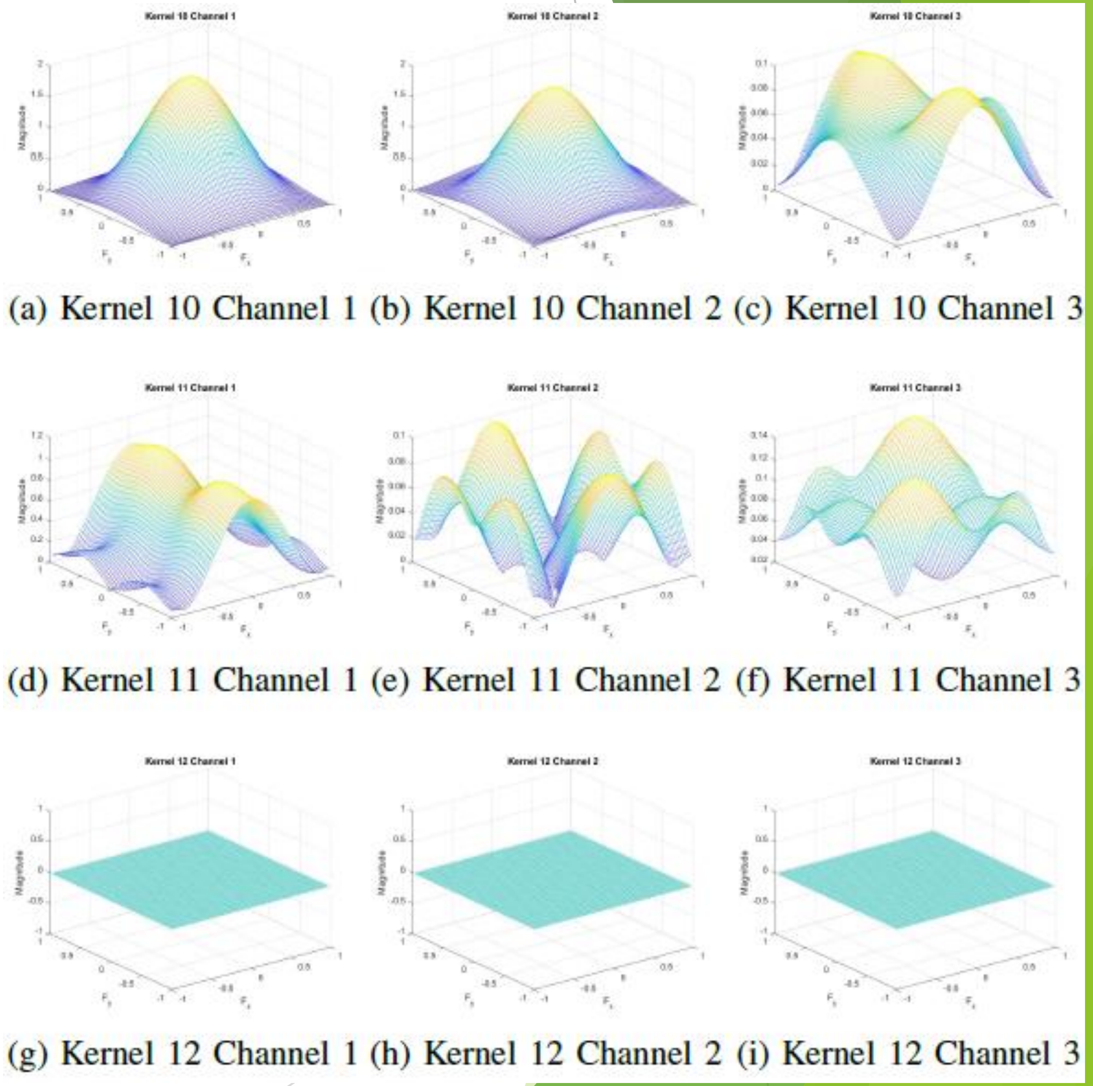
Training Dataset Samples

Fourier Transform Based Neural Network Pruning

- Goal: Make the network smaller and faster.
- We extract 3x3 convolution kernel weights and calculate their Fourier Transforms (FT).
- We remove similar filters according to their FT magnitudes.
- We notice that some kernels have nearly 0-response, so we remove them from the graph.
- We also eliminate those kernels with low energy at the 1x1 convolutional layers and the dense layer.







(a) Kernel 10 Channel 1 (b) Kernel 10 Channel 2 (c) Kernel 10 Channel 3

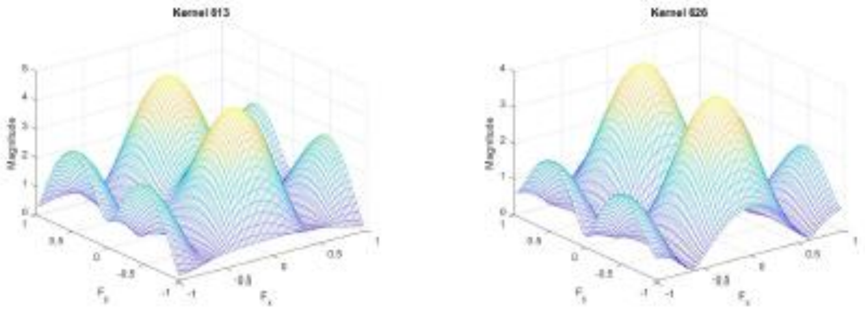
(d) Kernel 11 Channel 1 (e) Kernel 11 Channel 2 (f) Kernel 11 Channel 3

(g) Kernel 12 Channel 1 (h) Kernel 12 Channel 2 (i) Kernel 12 Channel 3

Frequency Response of the First Convolutional Layer

Fourier Transform Based Neural Network Pruning

- We perform 64x64 FT and check their cosine similarity:
- Filter similarity $= \cos(\theta) = \frac{<\mathbf{X},\mathbf{Y}>}{||\mathbf{X}||\cdot||\mathbf{Y}||}$ where $\mathbf{X}$ and $\mathbf{Y}$ are the Fourier transform magnitudes of the two filters in vector form, respectively.
- We treat the kernels with similarity larger than 0.99925 as a pair of similar kernels and store only one of them. This value is chosen based on no-fire video test experiment.
- In this way, 22.91% kernels are removed.

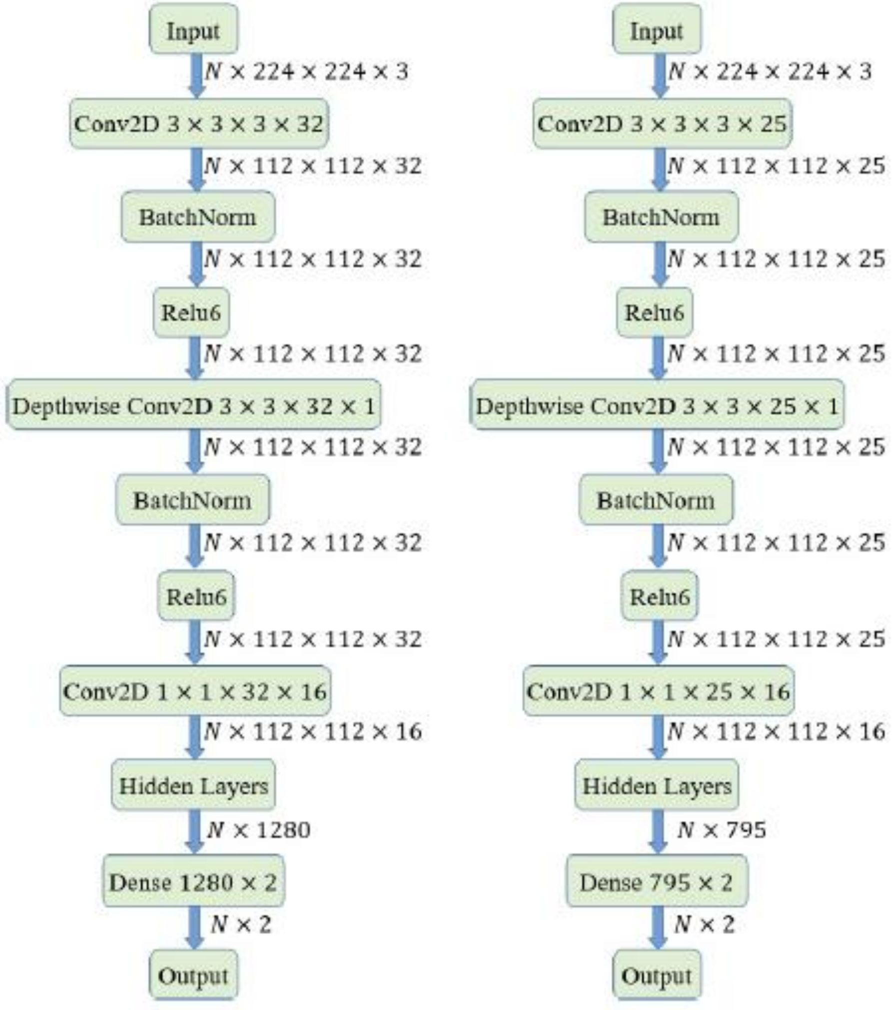


(a) Kernel 613 (b) Kernel 626

Frequency Response of the Final Convolutional Layer

Fourier Transform Based Neural Network Pruning

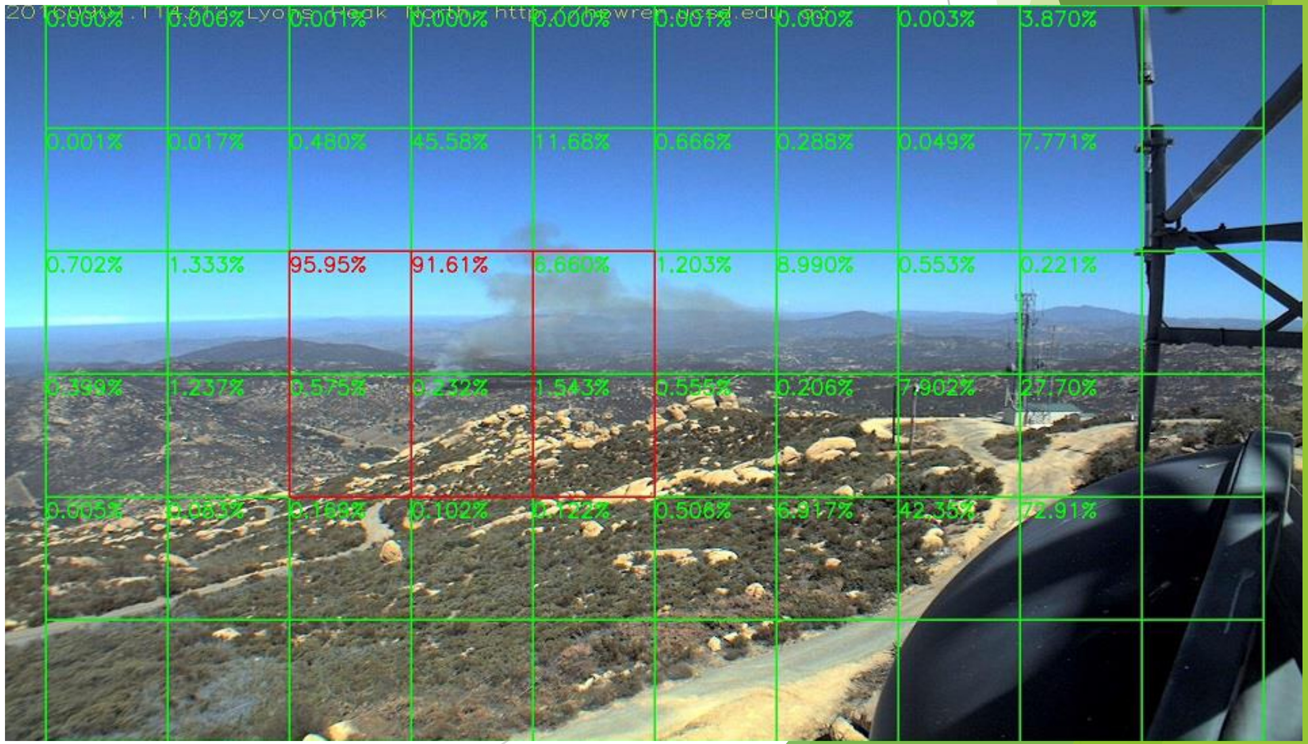
- A kernel with its shifted kernel may have a small cosine similarity in time-domain but their FT magnitudes will produce a $\cos(\theta) = 1$ or -1.
- By running the inference 100 times and calculate the average time consuming on the computer, we find that it takes 0.499921 second before pruning, and takes 0.464778 second after pruning.
- 7.04% time saving is achieved by pruning these layers.



(a) Before Pruning (b) After Pruning

Block-Based Image Frame Analysis

- Problem: In real time application, input frames are in 1080P or higher resolution, but the input of the network is 224x224. We may miss very small smoke regions, if we just down-sample the frames.
- We divide a frame into many small tiles.
- We have overlapping tiles.



Neural Network Performance: HPWREN

- The High Performance Wireless Research and Education Network (HPWREN), a University of California San Diego partnership project led by the San Diego Supercomputer Center and the Scripps Institution of Oceanography's Institute of Geophysics and Planetary Physics, supports Internet-data applications in the research, education, and public safety realms.
- Dataset: https://hpwren.ucsd.edu/

TABLE II: Daytime Fire Video Result of HPWREN Database

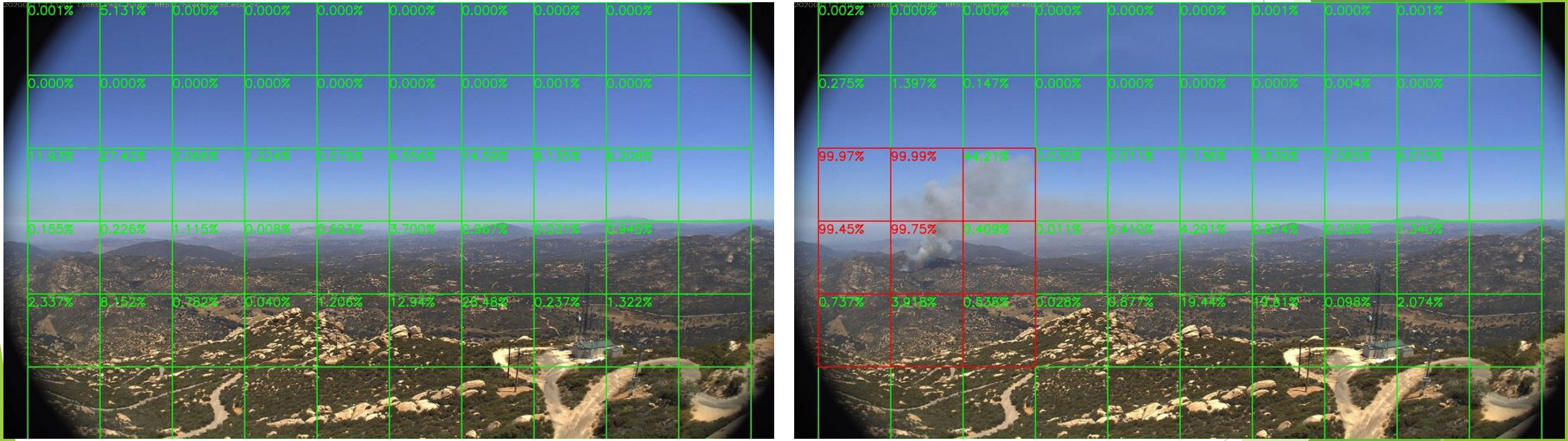
Videos Name	Fire Starts	First Detected		
		Fourier Domain Pruning	Time Domain Pruning	No Pruning
Lyons Fire	156	164	168	164
Holy Fire East View	721	732	738	732
Holy Fire South View	715	725	725	724
Skylinefire	684	690	690	690
Palisades Fire	636	639	640	639
Palomar Mountain Fire	262	277	279	275
Banner Fire	15	17	20	17
Highway Fire	4	6	6	6
DeLuz Fire	37	48	51	48

TABLE III: False-Alarm Result on No-Fire Videos of HPWREN Database

Threshold		None[a]		0.99925[a]		0.99900		0.99500	
Slimming Rate		0.00%		22.91%		24.90%		47.27%	
Videos Name	Frames	Num	Rate (%)	Num	Rate (%)	Num	Rate (%)	Num	Rate (%)
wilson-w-mobo-c[b]	10080	2	0.0198	2	0.0198	5	0.0496	63	0.6250
wilson-s-mobo-c	10074	2	0.0199	2	0.0199	2	0.0199	69	0.6849
wilson-n-mobo-c[b]	10024	3	0.0299	3	0.0299	4	0.0399	71	0.7083
wilson-e-mobo-c[c]	10028	43	0.4288	43	0.4288	43	0.4288	104	1.0371
vo-w-mobo-c	10009	5	0.0500	5	0.0500	5	0.0500	64	0.6394
69bravo-e-mobo-c	1432	1	0.0698	1	0.0698	1	0.0698	11	0.7682
69bravo-e-mobo-c	1432	0	0.0000	0	0.0000	0	0.0000	9	0.6285
syp-e-mobo-c	1421	3	0.2111	3	0.2111	3	0.2111	13	0.9148
sp-n-mobo-c	1252	2	0.1597	2	0.1597	2	0.1597	12	0.9585
sp-w-mobo-c[b]	1282	1	0.0780	1	0.0780	2	0.1560	8	0.6240
sp-s-mobo-c	1272	2	0.1572	2	0.1572	2	0.1572	8	0.6289
sp-e-mobo-c	1278	2	0.1565	2	0.1565	2	0.1565	10	0.7825

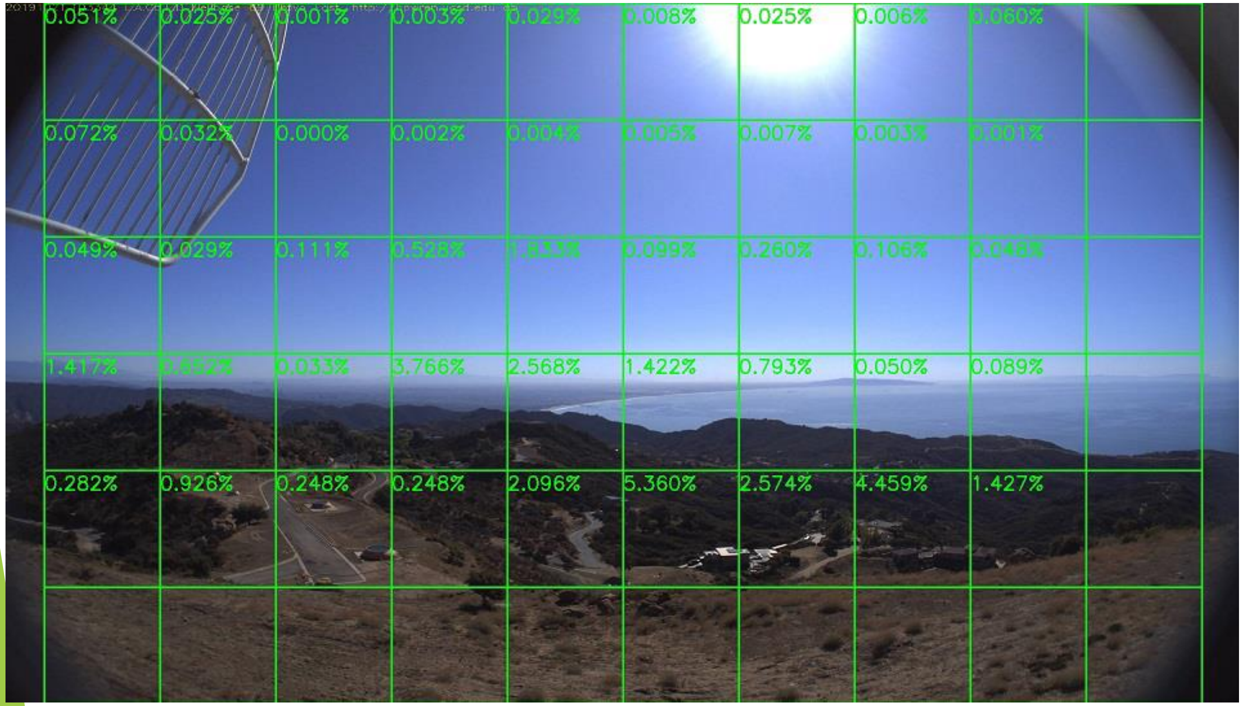
[a] We get same false-alarm result before and after pruning and slimming in threshold of 0.99925.
[b] With lower the slimming threshold (0.99900), the false-alarm rate increases on these videos.
[c] There is an unexpected long light shown in Fig.18b.

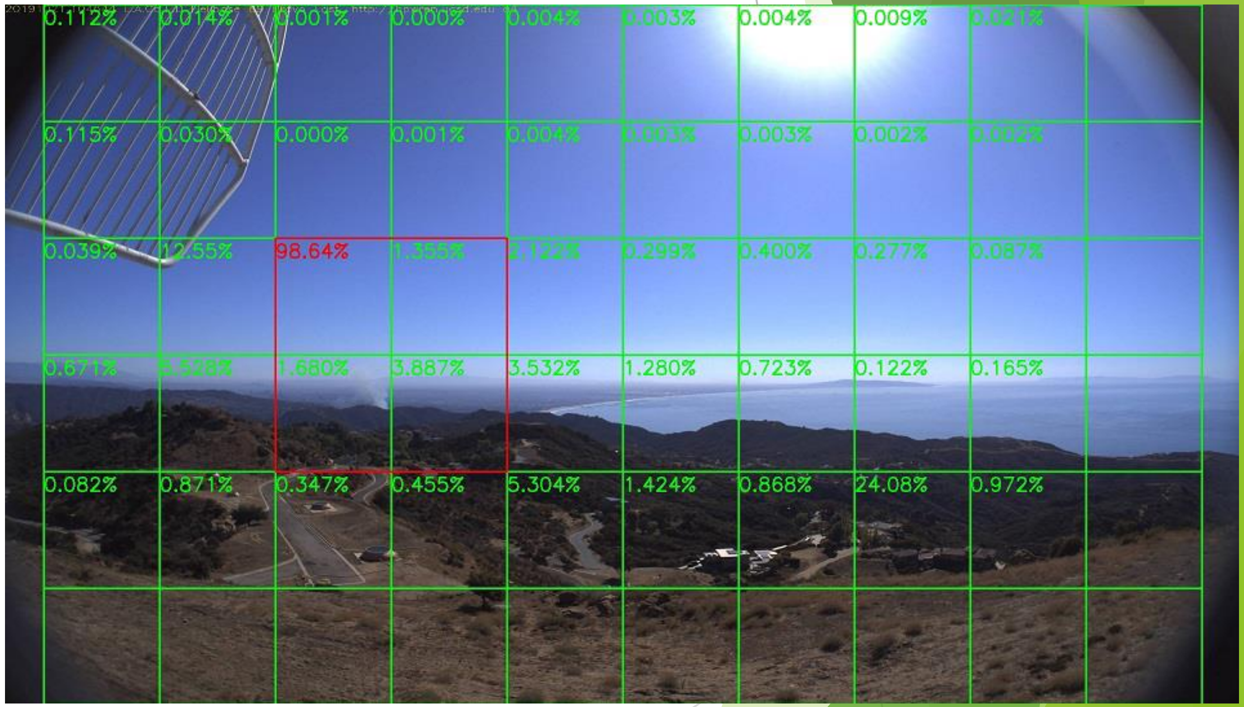
Neural Network Performance: HPWREN



Skylinefire north of Lyons Peak, 06/11/2020

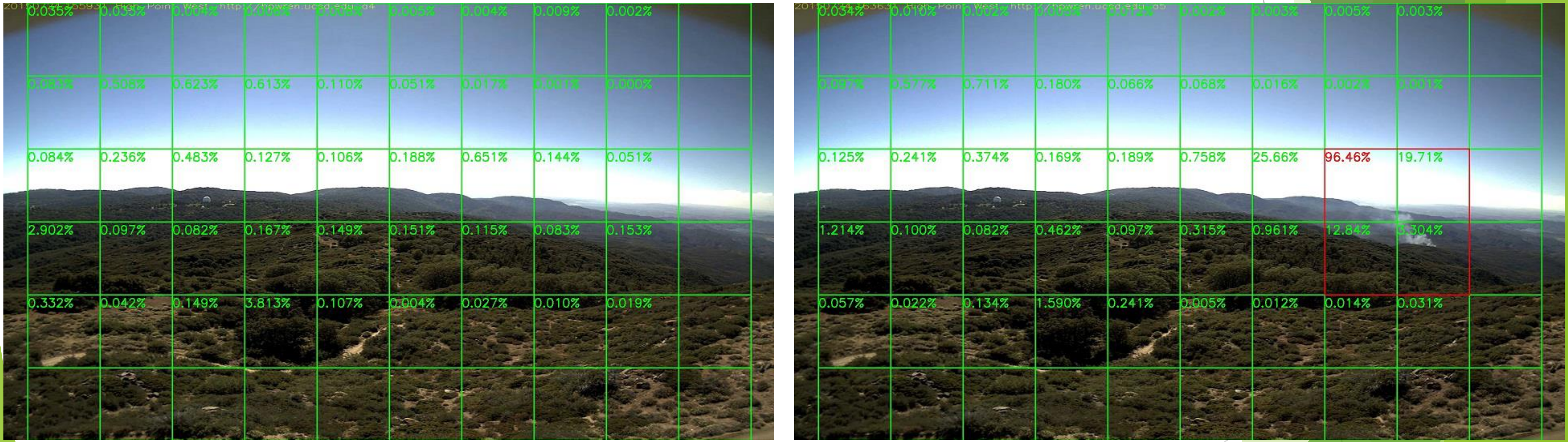
Neural Network Performance: HPWREN





Palisades fire, 10/21/2019

Neural Network Performance: HPWREN



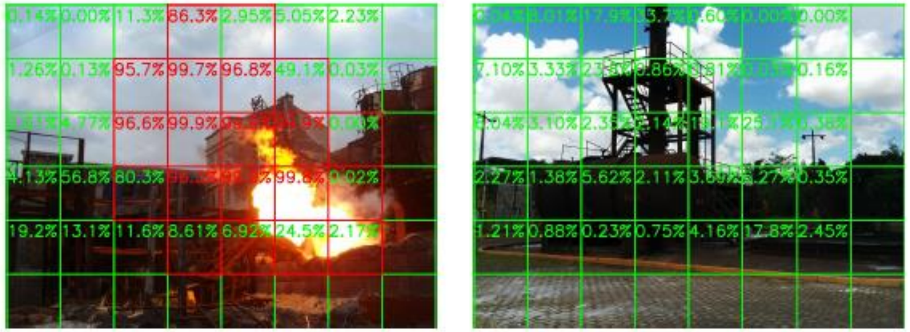
Palomar Mountain fire, 07/24/2015

Neural Network Performance: BoWFire

TABLE IV: Result of BoWFire Dataset

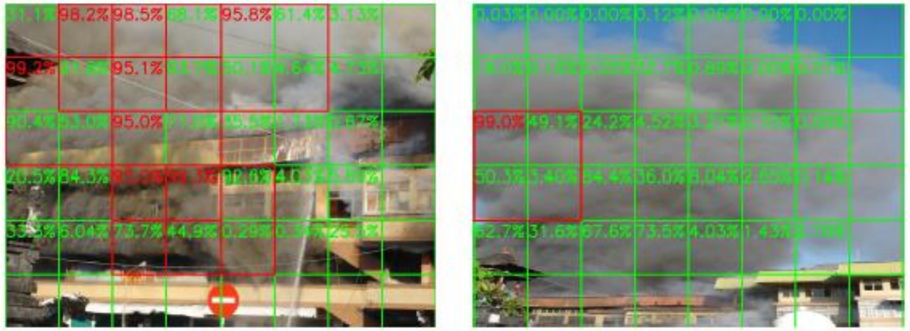
Method	Detection Rate	False-Alarm Rate	Accuracy
Muhammad et al. [30]	97.48%	18.69%	89.82%
Muhammad et al. [31]	93.28%	9.34%	92.04%
Chaoxia et al. [32]	92.44%	5.61%	93.36%
Our Method	91.60%	4.67%[a]	93.36%

[a] There are two smoke images labeled as no-fire as shown in Figure 20. Our method managed to detect them, but we still count them as false-alarm cases here for comparison because they are not discussed in [30]– [32]. If we count them as true-detected cases, then our three rates are 91.74%, 2.80% and 94.25%, respectively. Besides, as it is stated at the beginning of SectionIV-A, we sacrificed detection rate to gain false-alarm rate. Therefore, although our detection rate is lower than [30]– [32], our accuracy reaches the highest.



(a) Fire image No.34 (b) No-fire image No.98

Fig. 19: Test result on the BoWFire dataset.



(a) No-Fire Image No.63 (b) No-Fire Image No.64

Fig. 20: Two smoke images in the BoWFire no-fire test dataset.

Dataset: https://bitbucket.org/gbdi/bowfire-dataset/downloads/

Conclusion

- We trained a neural network for wildfire surveillance task via transfer learning.
- We pruned the network via Fourier Analysis.
- We use block-based image frame analysis to capture small smoke regions.
- We tested our system on HPWREN dataset and BoWFire dataset and obtained very good results.

- Thank your very much!
- Email: hpan21@uic.edu