Detecting LWIR Filters using Hyperspectral Camera and Neural Networks

Jiří Čech, Martin Rozkovec
Institute of Information Technologies and Electronics
FMMIS, Technical University of Liberec
Liberec, Czech Republic
jiri.cech@tul.cz, martin.rozkovec@tul.cz

Abstract **– We present detection of various filters using neural networks usable for our Long wave infrared (LWIR) hyperspectral detection system (HDES). Some reduction techniques are shown, for our aim of the small neural network with small computing requirements. In addition, the filter measurement is usable for calibration and verification of the HDES properties.**

Keywords-Hyperspectral imaging, LWIR, Neural networks, PCA

I. Introduction

Hyperspectral cameras achieve massive grow up especially in industrial applications, thanks to cheap bolometric sensors (FPA – Focal Plane Arrays) [1]. They are able to obtain image in short spectral range by the infrared camera and in combination with the special optics, we can provide very limited spectroscopy. The special optics breaks incoming radiation into very short spectral waveband and focuses it on the FPA. The spatial and spectral information is obtained from whole scene and it is stored in big three-dimensional data array called Hyperspectral cube (HC) [2]. Obtained HC needs to be transferred into the computation unit for data post processing, which usually have big computation requirements.

HDES works in LWIR band approximately from 7.5 to 11.5µm. It has up to 300px spectral resolution and 640px horizontal resolution with scanning mirror for vertical resolution (depends on the rotor of mirror). The optics of the HDES is on Figure 1. Hyperspectral image processing is possible to do at internal APSoC based control module Zynq 7020 [3] used as the system level controller and data acquisition device for the ULIS bolometric chip.

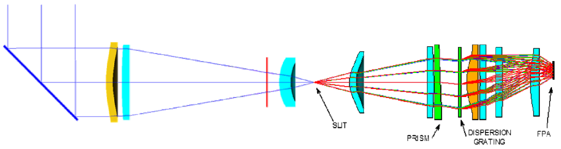

Figure 1. Optics of the HDES

The Zynq is programmable devices composed of two heterogeneous parts - Processing System (PS) and Programmable Logic (PL). PS is a multicore system on a chip (SoC) containing two ARM Cortex A9 32bit microprocessor cores running at 666MHz. Processors share 64KB L2 cache, interrupt controller, built-in peripheral controllers (IIC, SPI, gigabit Ethernet, USB, etc.), accesses to system memory (DDR3@1066MHz) and peripherals created in the PL. The PL part of the Zynq is a derivate of the Artix-7 family: low-power, low-cost branch of the 7-series Xilinx FPGAs.

Hyperspectral imaging usually utilizes two possible ways of the pixel spectral characteristics classification. The first case is called "Pure pixel" and takes place when a single material is classified. If it is a mixture of materials then it is called "Mixed pixel" classification. Both classification methods require different computational algorithms. With the neural network learning, we want use spectral and spatial information to achieve good results in both cases.

With big grow of computing power of PCs and other devices, simple or complex neural networks (NN) [4] are used as a common classifier in many fields like speech recognition, image quality enhancement or security. Usually a large dataset is required to train and to test an NN. To train a neural network one may use various programs, numerous plug-in modules for well known environments such as Matlab or libraries usable in different languages such as Python or .NET.

The structure of NN is based on layers. There are two mandatory layers – input and output one. Between them could be many hidden layers, which can provide some linear transformations, math operations (add bias, multiplication, mean, sum, log, etc.), reshaping, convolution, etc. There is an activation function between each two layers (ReLU, Tanh, etc.), that provides non-linear operation. The output layer is usually accompanied with transfer function that transforms results to log-probabilities.

Convolutional NNs (CNNs) [5] are commonly used in the image processing. CNN is a special type of a NN, which contains at least one convolutional layer. It is common, that this layer is accompanied with subsampling step. CNNs are able to detect shapes and objects. Their structure is composed of the convolutional preprocessing layer and a small fully connected NN. Size of CNN is smaller and it has lower number of hidden layers than common NN.

The convolution layers uses convolution with kernel of one, two or three dimensions and there are no more than three layers in a row. In front of them, there is a layer for size reduction such as subsampling layer or max pooling layer, which finds maximum value within the processed area. These operations are used to extract image features and use them as a reduced input fed into the NN. The example of CNN

ISBN 978-80-261-0642-5, © University of West Bohemia, 2017

with 5 convolutional layers for classify 224x224 pixel image into the 1000 categories, is shown on Figure 2.

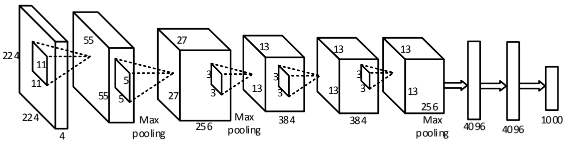

Figure 2. Example structure of CNN [5]

For the training of the created structure of NN, we need training data, which will be transformed into property input vector. Training is an iterative process. After each round, a difference between actual results and requested results is calculated. Based on these results, the selected criterial function updates values of weights in the NN. Then we evaluate actual NN on testing data set, which cannot be the same as the training data set. By repeating the training cycle, we want to minimize the error of testing dataset. This error is described as an accuracy of the NN.

II. Measurement

We present measurements for testing and calibration of the HDES by set of eight narrow bandpass filters and one neutral density filter with 50% transmittance. For each filter, we obtain HC with 300°C BlackBody (BB) [6] on the background. Data was post-process to mitigate invalid and corrupted pixels of the FPA with inbuilt correction of the infrared camera [7]. Also the non-uniformity correction [8] was applied, in order to unify non-linear characteristics of FPA pixels.

The measured data contain an inaccurate measurement of filters' transmittance. The accurate ones must be obtained with precise spectrometer only. The accuracy depends on the calibration and the thermal stability of the camera and on the environment conditions. The filters' transmittance is computed from measured data that are compared to BB plane, as it is shown in the Figure 3.

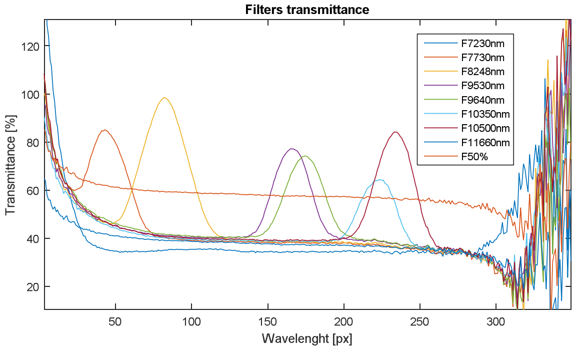

Figure 3. Obtained spectral characteristic of each filter (transmittance)

The data obtained from sensor provide valuable information up to the 300 px of the 480 px obtained from FPA row. We have cropped them out of HC for further post processing. To compare filters with real spectral characteristic, we can correct measured data, but to distinguish them, it is unnecessary to operate on raw data.

III. Neural Network

For our purposes (ie. NN training), we have selected the Torch [9], which is Lua language based framework, with wide support for machine learning algorithms. Also there is need to install additional modules as the NN, Optim, etc., based on user requirements.

Training of the NN is performed in batches and in each of them, we train some amount of the randomly chosen vectors from the training dataset. The randomness of chosen data minimizes the effect of errors and dependencies in training set. A testing cycle and output evaluation with criterial function is applied on randomly selected vectors. For that, we use Optim function with stochastic gradient descent optimization algorithm, which provides effective NN training.

The negative log likelihood criterion (NLL) is popular for NN training. It expects the log-probabilities of each class as its input, which can be obtained by adding the 'LogSoftMax' layer at the end of NN structure. We use cross entropy criterion, which is the same as NLL, but without the need of the NN structure modification.

For testing of NN we use the rest of source data, which was not use for training. The set of characteristic of each filter is good for pure pixel solution. For testing mixed pixel solution, we create mixture of the each two filters in ratio 90:10, 80:20, 70:30 and 60:40.

We use classic feed forward NN with hidden layers (if needed) for filter classification. Input vector has the same length, as number of wavelengths and output vector length is same as number of filters. We use ReLU as activation function, shown in the Equation 1. The results of this NN are shown in Table I.

$$x = max\ (0, x) \tag{1}$$

Used data was extracted from the measured HC from the selected polygon situated in the center of BB.

TABLE I. Results of classic feed forward NN

Hidden layers	0	1	2
Weights	2990	6180	9770
Data	**Successful [%]**		
Test 100	99.89	99.91	**99.96**
Test 90_10	99.75	**99.89**	98.76
Test 80_20	98.36	**98.92**	91.30
Test 70_30	**91.74**	91.24	75.21
Test 60_40	75.61	**75.69**	55.09

IV. Neural Network Reduction

We aim for the implementation of the recognizing spectral characteristic on FPGA, therefore we need small structure of the NN with the minimal weights. We compare three methods for input size reduction, like principal component analysis (PCA) [10], known input vector analyses and CNN.

The PCA is commonly used algorithm, which transforms input matrix into the set of uncorrelated vectors called principal components. Based on the variance of principal components we can minimize their number without loss of information (Figure 4) and their results are in Table II. Functions for counting PCA are available in Torch with module Unsup [11].

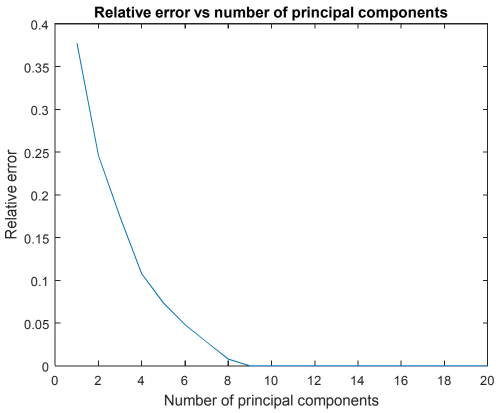

Figure 4. Relative error of PCA principal components reduction

TABLE II. RESULTS FOR PCA REDUCED INPUT VECTOR

Dimension	8	9	10	50
Data	**Successful [%]**			
Test 100	**99.94**	99.87	99.93	99.90
Test 90_10	75.45	85.39	80.75	**92.87**
Test 80_20	68.24	78.25	70.89	**90.45**
Test 70_30	64.64	68.02	58.88	**81.34**
Test 60_40	53.70	52.68	50.53	**65.03**

Because we want identify only a special set of characteristics, we can analyze source data and pick up just a few important wavelengths. The first method (MaxE) selects wavelengths with the maximal energy. The second method (Euclid) selects wavelengths with maximal mutual Euclidian distance compared to each other. The third method (Pearson) works similarly like Euclid, but it uses the Pearson correlation coefficient [13] for distance score evaluation. The wavelengths selected by the methods (pixels in source data) are shown in Figure 5 and their results are in Table III.

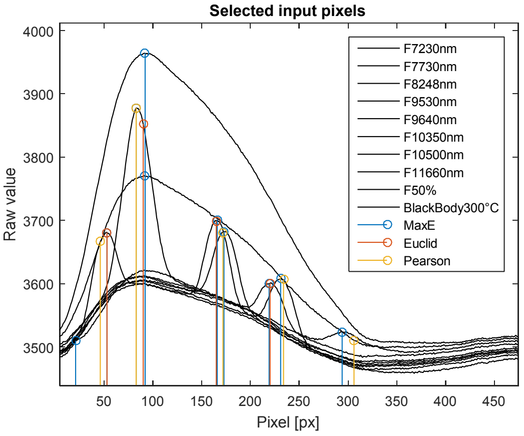

Figure 5. Selected values in source data

TABLE III. RESULTS FOR REDUCED INPUT VECTOR SIZE

Name	MaxE	Euclid	Pearson
Weights	90	40	50
Data	**Successful [%]**		
Test 100	98.99	91.02	**99.23**
Test 90_10	94.72	81.70	**95.47**
Test 80_20	**92.01**	76.89	87.07
Test 70_30	**82.14**	70.27	70.13
Test 60_40	**61.33**	54.59	52.93

We used measured spectral characteristics as the 1D input vector of CNN. We have used "Temporal convolution layer" (TCL) as the basic CNN layer, with following parameters: input frame size, output frame size, kernel width and step of the convolution (inputFrameSize, outputFrameSize, kernelWidth, stepWidth). It provides the convolution using the kernel in interval defined by stepWidth across input vector. The example of simple CNN is in model Conv1 with TCL (1,1,30,15) and linear layer (18,10).

The TCL is usually used with "Temporal max pooling layer" (TMPL) with size and step parameters (size, step), which shares input vector into the regions of defined size and selects from them only the maximal values. We used it in model Conv2 with TCL (1,2,30,20), TMPL (2,2) and linear layer (14,10) viz. Figure 6.

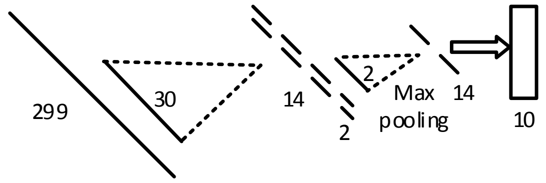

Figure 6. Structure of the model Conv2

We tried bigger structures of the CNN, as well. We employed two TCLs with different kernel sizes combined with TMPL. They are represented by the models Conv3 with TCL (1,2,30,5), TCL (2,2,4,2), TMPL (2,2), linear layer (26,10) and the model Conv4 with larger spacing in the first TCL (1,2,30,20), TCL (2,2,10,2), TMPL (2,2), linear layer (4,10). The results of tested CNN models are in Table IV.

TABLE IV. RESULTS OF THE CNN

Name	Conv1	Conv2	Conv3	Conv4
Weights	210	170	294	**80**
Data	**Successful [%]**			
Test 100	99.93	99.94	99.93	**99.99**
Test 90_10	**99.35**	96.52	98.78	94.91
Test 80_20	**95.60**	87.35	88.74	83.15
Test 70_30	**86.62**	77.37	67.67	66.60
Test 60_40	**66.12**	60.67	52.84	51.69

V. Conclusion

We used neural network classification for detection of the set of various LWIR filters. Due to their different spectral characteristic, the tested neural networks achieve a high success rate (98%) even with simple structure.

All created NNs were tested on the same dataset, which has pure pixel and mixed pixel characteristic of the filters. The NN for full input size contain almost 3000 weights and its classification success rate is the biggest in all of the aspects.

The method of reduced input size leads to rapid decrease of the NN size up to 40 weights. The Pearson reduction method reduced the size of NN to 50 weights and had better results than reduction by Euclidian method (Table III.).

With considering the data preprocessing, we can also minimize size of the NN. With the PCA method, we can reduce data input up to nine main components without loss of information, but the mixed pixel classification gains the worst results from the tested NNs (Table II.).

Using CNN reduces the number inputs to the regular NN. The size of NN with 210 weights provides the best results from the tested CNN (Table IV.). Compared to the big NN, which has average success rate 93% at the best, the CNN have 89.5% for pure and mixed pixel classification. The smallest tested CNN with size of 80 weights have 79.3% of average success rate.

On summarized Table V. we can see most successful cases from tested reduced NN. The best results have the CNN in all cases, but the methods of reduction input size leads to smaller amount of weights.

TABLE V. SUMMERIZED BEST RESULTS

Name	PCA50	MaxE	Conv1
Weights	500	**90**	210
Data	**Successful [%]**		
Test 100	99.90	98.99	**99.93**
Test 90_10	92.87	94.72	**99.35**
Test 80_20	90.45	92.01	**95.60**
Test 70_30	81.34	82.14	**86.62**
Test 60_40	65.03	61.33	**66.12**

Based on acquired results, we plan to use CNN with pre-selected ranges of input vector for common material detection. Use the spatial data of HC by exchange of 1D convolution to 3D convolution. Test the CNNs for dataset obtained with background BB with low temperature (ie. with high noise).

Acknowledgment

This work was partially supported by the Student Grant Scheme 2017 of the Technical University of Liberec, by the company APPLIC spol. s. r. o., and by the Ministry of Interior of the Czech Republic project no. VI20172020100.

References

[1] J.L. Tissot, P. Robert, A. Durand, S. Tinnes, E. Bercier and A. Crastes, "Status of uncooled infrared detector technology at ULIS", France, Defence Science Journal, vol. 63, no. 6, pp. 545-549, 2013.

[2] C. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis", IEEE Transactions on Information Theory, vol. 46, no. 5, pp. 1927-1932, 2000.

[3] Xilinx Inc., „Zynq-7000 All Programmable SoC", online <http://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>, 2017.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1, 2015.

[5] A. Geitgey, "Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks", online <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721#.c03a5knmc>, 2017.

[6] E. Theocharous, N.P. Fox, V.I. Sapritsky, S.N. Mekhontsev and S.P. Morozova, "Absolute measurements of black-body emitted radiance", Metrologia, vol. 35, no. 4, pp. 549-554, 1998.

[7] A.D.R. Girón and H.L. Correa, "A new algorithm for detecting and correcting bad pixels in infrared images", Ingenieria e Investigacion, vol. 30, no. 2, pp. 197-207, 2010.

[8] W. Isoz, T. Svensson and I. Renhorn, "Nonuniformity correction of infrared focal plane arrays", Proceedings of SPIE - The International Society for Optical Engineering, pp. 949, 2005.

[9] R. Collobert, C. Farabet, K. Kavukcuoglu and A. Chintala, "Torch", online<http://torch.ch>, 2017.

[10] H. Abdi and L.J. Williams, "Principal component analysis", Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, pp. 433-459, 2010.

[11] K. Kavukcuoglu, "Some unsupervised learning modules using Torch", online<https://github.com/koraykv/unsup>, 2017.

[12] X. Li, L. Jun, " An adaptive band selection algorithm for dimension reduction of hyperspectral images", Proceedings of 2009 International Conference on Image Analysis and Signal Processing, IASP 2009, pp. 114.

[13] S. Kaewpijit, J. Le Moigne and T. El-Ghazawi, "Automatic reduction of hyperspectral imagery using wavelet spectral analysis," in IEEE Transactions on Geoscience and Remote Sensing, vol. 41, no. 4, pp. 863-871, April 2003.