Approach and Conceptual Design

1. Why we choose Convolutional Neural Network?

Convolutional Neuron Network (CNN) is a highly efficient method, compared with other traditional methods like HOG (histogram of gradient) (Zhu et al., 2017). Because traditional methods use a dictionary to store all the human in the training data (Fischer, Herman, Behnke, & Systems, 2016). It has two drawbacks (Liu, Zhang, Wang, & Metaxas, 2016). Firstly, its speed is slow, because it compares with all the objects in the dictionary with the unknown object in the image. Secondly, if the algorithm sees a human which does not exist in the dictionary, the it is not able to detect it. In contrast, CNN uses feature extraction to capture the core of human shape.

2. What is Convolutional Neural Network?

CNN in object detection consists of two main processes: feature extraction and classification (Zhu et al., 2017). Feature extraction aims at extracting the distinctive features from images, such as a line, a curve, a tail and a nose. The more layers a CNN contains, the finer the extracted features will be. Classification means to compute the degree of certainty of whether an unknown object belongs to a certain category. If the degree of certainty is high, then the outcome will be it belongs to a class. An example of the CNN structure can be seen in the example of figure 1. Training a CNN means to refine and optimize the kernels (also called as weight or filter) such that the kernels are able to extract the most important and unique features that clearly distinguish each class from the rest of the classes.

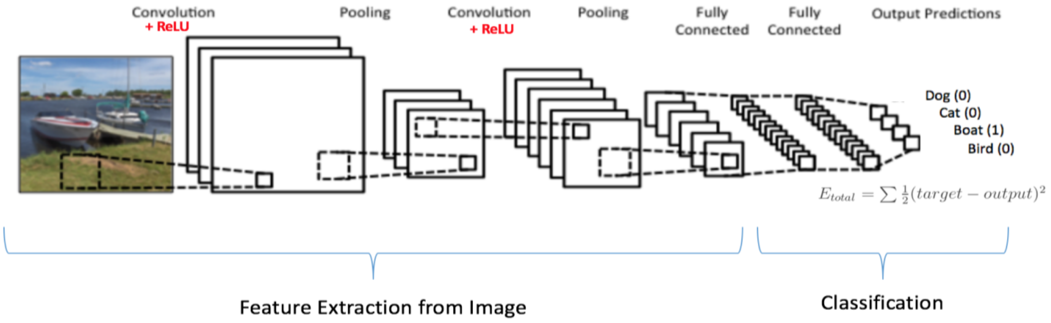

Figure 1 CNN architecture of detecting dog, cat, boat and bird from a visual image. CNN architecture contains feature extraction and classification.

2.1 feature extraction

The basic principle of feature extraction is to detect features from simple feature to complex feature(Hwang, Park, Kim, Choi, & So, n.d.). Simple feature contains edges and colours. Complex feature means a pattern that is assembled by multiple simple features. For example, in face recognition (figure 2), the first layer of CNN detects simple features, such as a vertical line, a horizontal line and an oblique line. The second layer detects middle level features, such as a nose, an eye or an ear. The final layer detects the most complicated feature which is the whole face in this case.

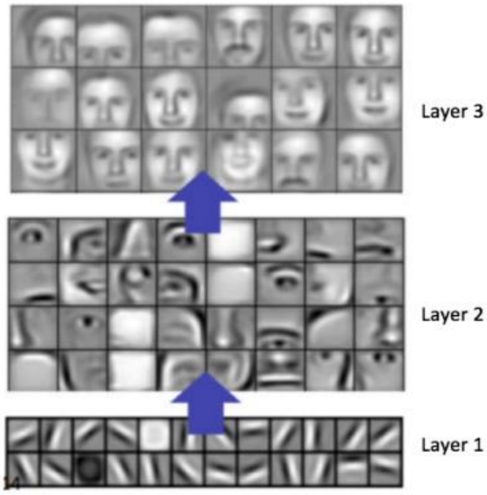

Figure 2 three layers of filters that are used to detect face features. Layer 1 aims at detecting simplest features. layer 2 aims at detecting the most complicated features

Feature extraction consists of four steps:

Step 1. **Convolutional Layer** (CONV): Convolutional layer applies filters on the input images. Filters are also called as kernels. Each kernel slides through the whole input image. A dot product of a kernel and a subset of the input image generates a pixel value in the output image(figure3). Different layer uses different kernels. In the layer of simple feature extraction, kernels are designed to detect lines and edges. In the layer of complex feature extraction, kernels are designed to detect complex patterns. The extraction of simple features contributes to the extraction of complex features. This can be seen in the hierarchy (figure 4).

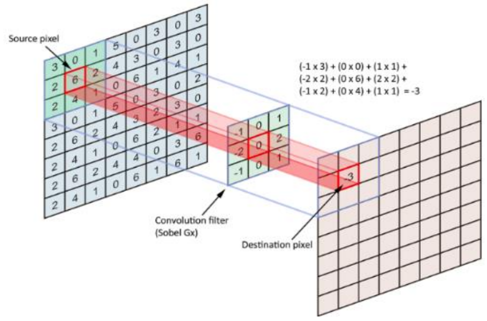
Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel

Figure 3 dot product of a kernel and a subset of image. The kernel slides through the whole image

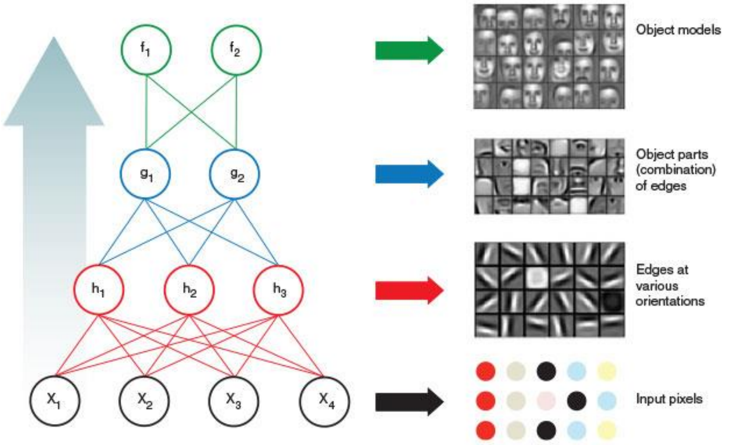
f1
f2
g1
g2
h1
h2
h3
X1
X2
X3
X4
Object models
Object parts (combination) of edges
Edges at various orientations
Input pixels

Figure 4 a hierarchy that shows the extraction of simple features (e.g.red neurons) contributes to the extraction of complex features (e.g. blue neurons)

Step 2. **Rectifying Linear Units Layer** (RELU): In order to remove negative values in the output images of a convolutional layer, a threshold is applied.

Step 3. **Maximum Pooling Layer** (POOL)**:** The goal of maximum pooling layer is to decrease the size of the output images of each convolutional layer and do not distort them. Therefore, the speed of further computation will not slow down.

Step 4. **Iterate the above 3 steps.** The more amount of iterations there are, the finer the features will be extracted.

2.2 Classification

The classification process calculates the degree of certainty that an object belongs to a class (Hwang et al., n.d.). The classification is based on the output of feature extraction. For instance, boat body is one of the features in the output of feature extraction. If the presence of boat body is clearly extracted, then it is unlikely that this is a dog or a cat or a bird, because they do not have a boat body. In the example of figure 6, the probability that this object is a boat is 0.94 (figure 5). Therefore, the output is that this is a boat.

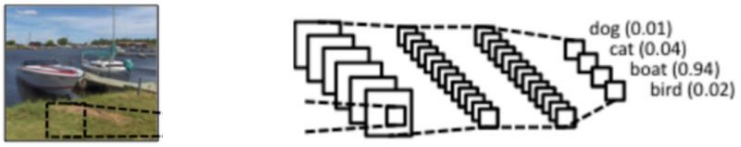

Figure 5 The left image is an input image of CNN. The right image shows that, after feature extraction, the classification process gives the output: The probability that an object in this image is a dog, a cat, a boat and a bird is 0.01, 0.04, 0.94 and 0.02 respectively.

3. Workflow Of This Project

Thermal images and visual images will be integrated in CNN for human detection. Because these two types of images provide complementary detection decisions. When illumination is sufficient, such as in a daytime, it is easy to distinguish objects by visual images. When illumination is insufficient, such as during the night or in a bad weather, thermal images have a better performance. So the integration of visual and thermal images will provide the most accurate detection, compared with using one of them. The workflow of this project is the following.

1) Data collection: Thermal images and visual images will be collected from several conflict areas which has a similar type of conflict as Lebanon, a similar terrain as Lebanon and a similar urban infrastructure as Lebanon.
2) Data annotation: Human in both of these two stacks of images will be labeled (also called annotated) manually. A label is a bounding box of a human. These labels are seen as ground truth.
3) Data splitting: The labeled thermal images and visual images will be splitted into two groups. One group is the training data. The other group is the test data.
4) Training a CNN:

   The training process contain 5 steps.

   ① Input of CNN: Thermal images and visual images will be the input layer.

   ② Feature Extraction: Feature will be extracted from simple ones (edges and lines), to complicated ones (an arm, a leg, a foot, a head), to more complicated ones (upper body and lower body) and finally to the whole human body. The process is expected to be very similar as figure 6.

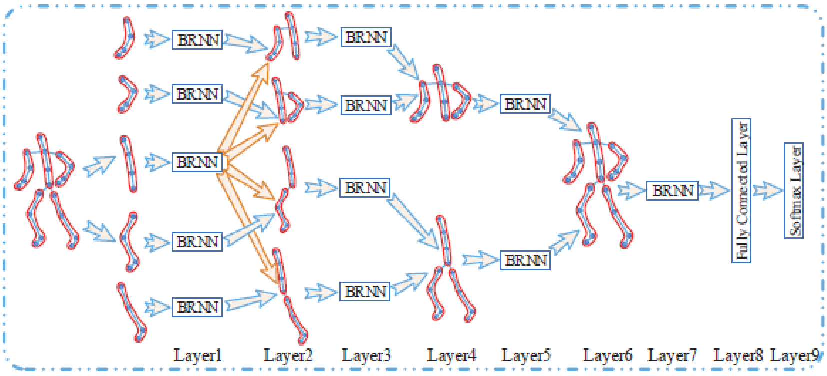

Figure 6 CNN architecture of human detection. The image on the most left hand side is an input image. Features visualized with red contours are the features that each layer aims at extracting. For example, layer 1 extracts an arm. Layer 2 extracts an upper body with an arm. Layer 4 extracts an upper body with two arms. Layer 6 extract the whole human body.

   ③ Classification: The probability of there is a human or not on a certain location will be calculated.

   ④ Output: Make a conclusion whether there is human or not on each location.

   ⑤ Draw a bounding box on each detected person in visual images and thermal images (figure 15 and 16).

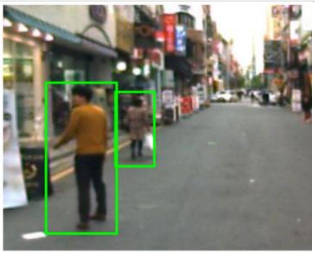

Figure 15 bounding box on an visual image

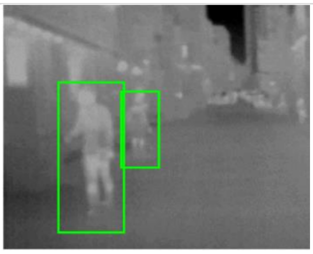

Figure 16 bounding box on a thermal image

5) Testing of CNN: After the CNN has been trained, it will be tested. The bounding box drawn by CNN will be compared with the bounding box drawn in ground truth. The requirement of the accuracy will be made based on the agreement of us and our user.
6) Applying CNN to Lebanon conflict area: If the accuracy satisfies the requirement, the trained CNN will be applied to conflict area.

Reference

Fischer, V., Herman, M., Behnke, S., & Systems, A. I. (2016). Multispectral Pedestrian Detection using Deep Fusion Convolutional Neural Networks, (April), 27–29.

Hwang, S., Park, J., Kim, N., Choi, Y., & So, I. (n.d.). Multispectral Pedestrian Detection : Benchmark Dataset and Baseline.

Liu, J., Zhang, S., Wang, S., & Metaxas, D. N. (2016). Multispectral Deep Neural Networks for Pedestrian Detection, 1–13. Retrieved from http://arxiv.org/abs/1611.02644

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8–36. https://doi.org/10.1109/MGRS.2017.2762307