Introduction to Computer Vision

Classwork – Paper Summarization

Deep Learning vs. Traditional Computer Vision

Lecturer: **Dr. Pham Van Huy**
Student name: **Do Pham Quang Hung**
Student ID: **520K0127**

Deep Learning vs. Traditional Computer Vision

Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli,
Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, Joseph Walsh

IMaR Technology Gateway, Institute of Technology Tralee, Tralee, Ireland
niall.omahony@research.ittralee.ie

Contents

I. Introduction

Deep Learning (DL) is used in the domain of digital image processing to solve difficult problems (e.g., image colourization, classification, segmentation and detection). DL methods such as Convolutional Neural Networks (CNNs) mostly improve prediction performance using big data and plentiful computing resources and have pushed the boundaries of what was possible. Problems which were assumed to be unsolvable are now being solved with super-human accuracy. Image classification is a prime example of this. Since being reignited by Krizhevsky, Sutskever and Hinton in 2012 [1], DL has dominated the domain ever since due to a substantially better performance compared to traditional methods.

This paper will provide a comparison of deep learning to the more traditional hand-crafted feature definition approaches which dominated CV prior to it. There has been so much progress in Deep Learning in recent years that it is impossible for this paper to capture the many facets and sub-domains of Deep Learning which are tackling the most pertinent problems in CV today. This paper will review traditional algorithmic approaches in CV, and more particularly, the applications in which they have been used as an adequate substitute for DL, to complement DL and to tackle problems DL cannot.

II. A Comparison of Deep Learning and Traditional Computer Vision

2.1. What is Deep Learning?

DL is a subset of machine learning. DL is based largely on Artificial Neural Networks (ANNs), a computing paradigm inspired by the functioning of the human brain. Like the human brain, it is composed of many computing cells or 'neurons' that each perform a simple operation and interact with each other to make a decision [6]. Deep Learning is all about learning or 'credit assignment' across many layers of a neural network accurately, efficiently and without supervision and is of recent interest due to enabling advancements in processing hardware [7]. Self-organisation and the exploitation of interactions between small units have proven to perform better than central control, particularly for complex non-linear process models in that better fault tolerance and adaptability to new data is achievable [7].

2.2. Advantages of Deep Learning

Rapid progressions in DL and improvements in device capabilities including **computing power, memory capacity, power consumption, image sensor resolution, and optics have improved the performance and cost-effectiveness** of further quickened the spread of vision-based applications. Compared to traditional CV techniques, DL enables CV engineers to achieve **greater accuracy** in tasks such as image classification, semantic segmentation, object detection and Simultaneous Localization and Mapping (SLAM). Since neural networks used in DL are trained rather than programmed, applications using this approach often require less expert analysis and fine-tuning and exploit the tremendous amount of video data available in today's systems. DL also provides **superior flexibility** because CNN models and frameworks can be re-trained using a custom dataset for any use case, contrary to CV algorithms, which tend to be more domain-specific.

Taking the problem of object detection on a mobile robot as an example, we can compare the two types of algorithms for computer vision:

- The *traditional* approach is to use well-established CV techniques such as **feature descriptors (SIFT, SURF, BRIEF, etc.)** for object detection. Before the emergence of DL, a step called feature extraction was carried out for tasks such as image classification. **Features are small "interesting", descriptive or informative patches in images**. Several CV algorithms, such as edge detection, corner detection or threshold segmentation may be involved in this step. As many features as practicable are extracted from images and these features form a definition (known as a bag-of-words) of each object class. At the deployment stage, these **definitions are searched** for in other images. If a significant number of features from one bag-of-words are in another image, the image is classified as containing that specific object (i.e. chair, horse, etc.)
- *Deep Learning* introduced the concept of **end-to-end** learning where the machine is just given a dataset of images which have been annotated with what classes of object are present in each image [7]. Thereby a DL model is 'trained' on the given data, where **neural networks discover the underlying patterns in classes of images and automatically works out the most descriptive and salient features with respect to each specific class of objec**t for each object. It has been well-established that DNNs perform far better than traditional algorithms, albeit with trade-offs with respect to computing requirements and training time. With all the state-of-the-art approaches in CV employing this methodology, the **workflow** of the CV engineer **has changed dramatically** where the knowledge and expertise in extracting hand-crafted features has been replaced by knowledge and expertise in iterating through deep learning architectures as depicted in Fig. 1.

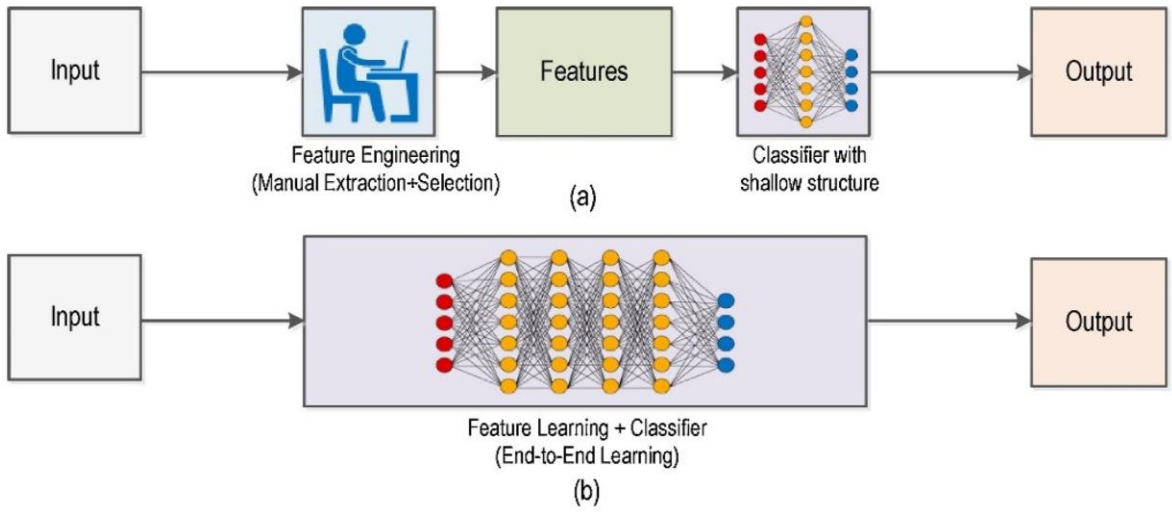

Fig. 1. (a) Traditional Computer Vision workflow vs. (b) Deep Learning workflow. Figure from [8].

The development of CNNs has had a tremendous influence in the field of CV in recent years and is responsible for a big jump in the ability to recognize objects. **CNNs make use of kernels (also known as filters), to detect features (e.g. edges) throughout an image. A kernel is just a matrix of values, called weights, which are trained to detect specific features**. As their name indicates, the main idea behind the CNNs is to spatially convolve the kernel on a given input image check if the feature it is meant to detect is present. To provide a value representing how confident it is that a specific feature is present, *a convolution operation is carried out by computing the dot product of the kernel and the input area where kernel is overlapped* (the area of the original image the kernel is looking at is known as the receptive field [10]).

To facilitate the learning of kernel weights, the **convolution layer's output is summed with a bias term** and then **fed to a non-linear activation function**. Activation Functions are usually non-linear functions like Sigmoid, TanH and ReLU (Rectified Linear Unit). Depending on the nature of data and classification tasks, these activation functions are selected accordingly [11]. For example, ReLUs are known to have more biological representation (neurons in the brain either fire or they don't). As a result, *it yields favourable results for image recognition tasks as it is less susceptible to the vanishing gradient problem and it produces sparser, more efficient representations*

To speed up the training process and reduce the amount of memory consumed by the network, the convolutional layer is often followed by a pooling layer to remove redundancy present in the input feature.

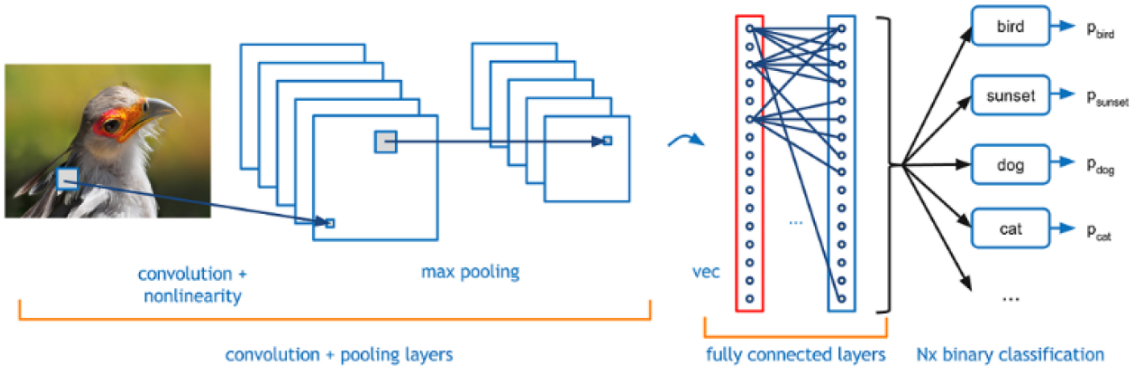

Fig. 2. Building blocks of a CNN.

2.3. Advantages of Traditional Computer Vision Techniques

This section will detail how the traditional feature-based approaches such as those listed below have been shown to be useful in improving performance in CV tasks:

- Scale Invariant Feature Transform (SIFT) [14]
- Speeded Up Robust Features (SURF) [15]
- Features from Accelerated Segment Test (FAST) [16]
- Hough transforms [17]
- Geometric hashing [18]

Feature descriptors such as SIFT and SURF are generally combined with traditional machine learning classification algorithms such as **Support Vector Machines** and **K-Nearest Neighbours** to solve the aforementioned CV problems.

DL is sometimes overkilling as often traditional **CV techniques can solve a problem much more efficiently and in fewer lines of code than DL**. Algorithms like SIFT and even simple colour thresholding and pixel counting algorithms are not class-specific, that is, they are very general and perform the same for any image. In contrast, **features learned from a deep neural net are specific to your training dataset** which**, if not well constructed, probably won't perform well for images different from the training set**. Therefore, SIFT and other algorithms are often used for applications such as image stitching/3D mesh reconstruction which don't require specific class knowledge. These tasks have been shown to be achievable by training large datasets, however this requires a huge research effort and it is not practical to go through this effort for a closed application. One needs to practice common sense when it comes to choosing which route to take for a given CV application. For example, to classify two classes of product on an assembly line

conveyor belt, one with red paint and one with blue paint. A deep neural net will work given that enough data can be collected to train from. However, the same can be *achieved by using simple colour thresholding*. Some **problems can be tackled with simpler and faster techniques**.

What if a DNN performs poorly outside of the training data? **If the training dataset is limited, then the machine may overfit to the training data and not be able to generalize for the task at hand**. It would be too difficult to manually tweak the parameters of the model because a DNN has millions of parameters inside of it each with complex inter-relationships. In this way, DL models have been criticised to be a black box in this way [5]. **Traditional CV has full transparency and the one can judge whether your solution will work outside of a training environment**. The CV engineer can have insights into a problem that they can transfer to their algorithm and if anything fails, the parameters can be tweaked to perform well for a wider range of images.

Today, the traditional techniques are used when the **problem can be simplified so that they can be deployed on low-cost microcontrollers** or to limit the problem for deep learning techniques by highlighting certain features in data, augmenting data [19] or aiding in dataset annotation [20]. We will discuss later in this paper how many image transformation techniques can be used to improve your neural net training. Finally, there are many more challenging problems in CV such as: Robotics [21], augmented reality [22], automatic panorama stitching [23], virtual reality [24], 3D modelling [24], motion estimation [24], video stabilization [21], motion capture [24], video processing [21] and scene understanding [25] which cannot simply be easily implemented in a differentiable manner with deep learning but benefit from solutions using "traditional" techniques

Deep Learning	Traditional Computer Vision
Complex NN architecture	Simpler solution
Ability to handle noisy, incomplete or unstructured data.	Work well on limited given data
Ability to learn from large amounts of data without much human intervention or prior knowledge	Can be deployed on low-cost devices
Higher accuracy and versatility for complex tasks like object detection, face recognition, natural language processing	Faster inference time, shorter training time
	Transparency and interpretability of the models and their decisions