# **What is the state of AI in computer vision?**

With the growing power of our computers, most of the applications like image classification, image segmentation, object detection, and more complex tasks like face recognition, image generation, or even style transfer applications are now being referred to as “artificial intelligence models”. The major difference is our computation power coming from the recent advancements of GPUs.

#### **Convolutional Neural Networks:**

convolution is a process where an original image, or video frame, which is our input in a computer vision application, is convolved using filters that detect important small features of an image such as edges. The network will autonomously learn filter values that detect important features to match the output we want to have, such as the object’s name in a specific image sent as input for a classification task.

**Deep Nets  — Strengths and Weaknesses:**

What is the future of computer vision algorithms? As you may be thinking, one way to improve computer vision applications is to understand our own visual system better, starting with our brain, which is why neuroscience is such an important field for AI. Indeed, current deep nets are surprisingly different from our own vision system. Firstly, humans can learn from minimal numbers of examples by exploiting our memory and the knowledge we already acquired. We can also exploit our understanding of the world and its physical properties to make deductions, which a deep net cannot do.

Deep nets also have strengths that we must highlight. They can outperform us for face recognition tasks since humans are not used to, until recently, seeing more than a few thousand people in their whole lifetime. But this strength of deep nets also comes with a limitation where these faces need to be straight, centered, clear, without any occlusions, etc.

#### **Deep Nets vs. the Human Vision System:**

This dataset limitation comes with the price that these deep neural networks are much less general-purpose, flexible, and adaptive than our own visual system. They are less general-purpose and flexible in the way that, contrary to our visual system where we automatically perform edge detection, binocular stereo, semantic segmentation, object classification, scene classification, and 3D depth estimation, deep nets can only be trained to achieve one of these tasks. Indeed, simply by looking around, your vision system automatically achieves all these tasks with extreme precision where deep nets have difficulty achieving similar precision on one of them. But even if this seems effortless to us, half of our neurons are at work processing the information and analyzing what is going on.

One of the biggest limitations of deep nets is that they are dependent on data. Indeed, the lack of precision we previously mentioned by deep nets arises mainly because of the disparity between the data we use to train our algorithm and what it sees in real life. As you know, an algorithm needs to see a lot of data to iteratively improve at the task it is trained for. This data is often referred to as a training dataset.

#### **The data dependency problem:**

This data disparity between the training dataset and the real world is a problem because the real world is too complicated to accurately be represented in a single dataset, which is why deep nets are less adaptive than our vision system called as the combinatorial complexity explosion of natural images. The combinatorial complexity comes from the multitude of possible variations within a natural image like the camera pose, lighting, texture, material, background, the position of the objects, etc. Biases can appear at any of these levels of complexity the dataset is missing.

#### **Benchmarks:**

Currently, we use benchmarks with the most complex datasets possible to compare the current algorithms and rate them, which, if you recall, are incomplete compared to the real world. Nonetheless, we are often happy with 99% accuracy for a task on such benchmarks. Firstly, the problem is that this 1% error is determined on a benchmark dataset, meaning that it is similar to our training dataset in the way that it does not represent the richness of natural images. It’s normal because it is impossible to represent the real world in just a bunch of images, it is too complicated, and there are too many situations possible. These benchmarks we use to test our dataset to determine whether or not they are ready to be deployed in the real-world application are not really accurate to determine how well it will \*actually\* perform, which leads to the second problem that is how it will actually perform in the real world.

Let’s say that the benchmark dataset is huge and most cases are covered, and we really have 99% accuracy. What are the consequences of the 1% of cases where the algorithm fails in the real world? This number will be represented in misdiagnosis, accidents, financial mistakes, or even worse, deaths.

Such cases could be a self-driving car during a heavy rainy day, heavily affecting the depth sensors used by the vehicle, causing it to fail many depth estimations. Would you trust your life to this partially-blind “robotaxi”?

I don’t think I would. Similarly, would you trust a self-driving car at night to avoid driving over pedestrians or cyclists where even yourself had difficulty seeing them? These kinds of life-threatening situations are so broad that it’s almost impossible that they are all represented in the training dataset.

Especially now, where most applications are made for real-life uses instead of only academic competitions, it is crucial to get out of these *academia evaluation metrics* and create more appropriate evaluation tools. We also have to accept that data bias exists and that it can cause real-world problems. Of course, we need to learn to reduce these biases but also to accept them. Biases are inevitable due to the combinatorial complexity of the real world that cannot be realistically represented in a single dataset of images yet. Thus, focusing our attention, without any play on words with transformers, on better algorithms that can learn to be fair even when trained on such “incomplete” datasets rather than having bigger and bigger models trying to represent the most data possible.