Deep Convolutional neural network for Fingerprint Recognition

Elham Tabassi Xiao Zeng

1. Introduction

Fingerprints are ridge and valley patterns present on the surface of human fingertips. Fingerprint matching has played a critical role in identifying suspects and criminals. Due to fingerprints high discriminability and persistence over time, fingerprint-based person recognition systems have been widely deployed. Fingerprint recognition systems have played a crucial role in various applications, including law enforcement, forensics, physical and logical access control, border crossing and civil registry. Fingerprint features can be divided into three major categories based on the granularity at which they are extracted: level 1, level 2 and level 3 features. In this project, we are going to implment a fingerprint recognition system that classify a fingerprin into the following 5 categories: A=Arch, L=Left Loop, R=Right Loop, T=Tented Arch, W=Whorl. As shown in Fig.1

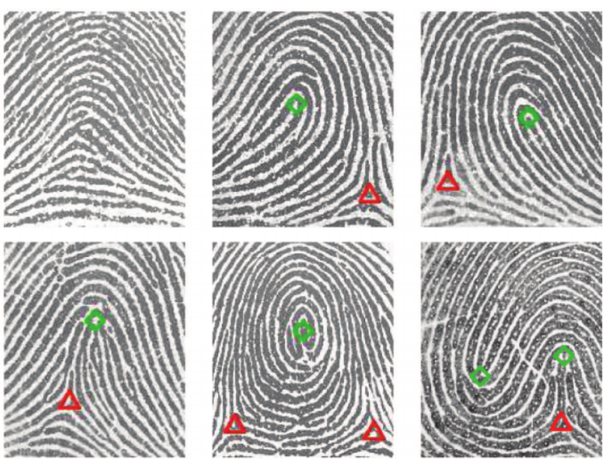

Figure 1. Fingerprint classes: Top row: arch, left loop, and right loop. Bottom row: tented arch, whorl, and twin-loop. Triangles mark deltas; diamonds mark cores and whorls. Note that there is an invisible delta further to the bottom left of the last image, which actually shows a tented-arch-and-loop rather than a twin-loop. These are based on images from the FVC 2000, Database 2a [34].

From the last image in Fig.1 we can see that one fingerprint may be classified into multiple categories at the same time.

2. Methodology

2.1. Deep Learning

In this project, we plan to develop and implement a novel deep learning algorithm for fingerprint claasifications. Specifically, we will implement some state-of-the-art convolutional neural network (CNN) architecture as baselines, such as GoogLeNet [1]. Our goal is to propose a novel approach that tries to devise some novel CNN architecture that aims at classifying fingerprints.

3. Experiments

In this section, we empirically evaluate the effectiveness and the efficiency of the state-of-arts in deep reinforcement learing on OpenAI gym Atari environment. Note that at current stage, we didn't propose any new methods in deep reinforcement learning; instead our first stage is to learning the most advanced algorithms first using public available online resources.

3.1. Environments

In order to evaluate the algorithm, we conduct our experiments on Atari environment provided by OpenAI. The description of Atari environment is as follows:

> Maximize your score in the Atari 2600 game. In this environment, the observation is an RGB image of the screen, which is an array of shape (210, 160, 3) Each action is repeatedly performed for a duration of kk frames, where kk is uniformly sampled from $\{2, 3, 4\}$ [**?**]

We choose three Atari games as our testing environment, all of which are unsolved environments, which means those games don't have a specific reward that you can consider it as the end of game:

1. Breakout-v0
   In this game, player control a paddles at the bottom of screen and try to bounce the ball upwards to hit those bricks as soon and as much as possible, as illustrated in Fig **??** (a).

2. Pong-v0
   In this game, players control a paddles at the right of

screen and try to bounce the ball pass the other player at the left of screen, as illustrated in Fig **??** (b).

3. Phoenix-v0 In this game, players control a spaceship by moving it horizontally at the bottom of screen, as illustrated in Fig **??** (c), trying to destroy the enemies by firing upwards and avoiding the attack from those enemies.

3.2. Baselines

In this section, we introduce the implementation details of the baselines we used.

The structure of policy network is shown as in Table.1:

Table 1. CNN detail

Type	**size**	**# filters**	**activation**
convolution	5×5	32	Relu
max pooling	2×2		
convolution	5×5	32	Relu
max pooling	2×2		
convolution	4×4	64	Relu
max pooling	2×2		
convolution	3×3	64	Relu
fully connected	512		PRelu
softmax			

We modified the implementation of A3C from tensorpack repository.

4. Time Line

10/19-10/26: Review some related literatures about deep reinforcement learning-based games and related deep reinforcement learning methods. Figure out possible suitable methodologies and games to implement.

10/27-11/02: Review paper "Asynchronous Methods for Deep Reinforcement Learning". Learn to use OpenAI Gym, Keras software package.

11/09-11/16: Use Keras to define the deep q network. Write midterm paper.

11/16-11/23: OpenAI's gym library to interact with the game Learning Environment.

11/23-11/30: Use Tensorflow to optimization the network

12/01-12/13: Test the game. Write final report.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.