Rethinking Offline Personalized Advertising: Challenges and System Design

Ashutosh Sathe
Department of Computer Engineering
College of Engineering, Pune
Pune, India, 411005

Sunil B. Mane
Department of Computer Engineering
College of Engineering, Pune
Pune, India, 411005

ABSTRACT

Online personalized advertisements on a smartphone have shown a great impact on both user experience and advertiser income. This type of personalized advertising is possible due to existence of easily traceable features when user is online. Online advertising agencies such as Google AdSense can determine appropriate ads for a particular user based on their behavior on the internet. Therefore, these advertising agencies inherently depend long interactions between user and the device to get a decent ad recommendation. This paper focuses on interactions of users with electronic devices which are very short and need-based. Examples of these devices would be gaming arenas, selfie stations in malls or self check-in booths at the airport. The paper throughout considers these types of interactions as ”offline” since there is no way to track user’s behavior here like it is possible in ”online” scenario. Main objective of the paper is to discuss challenges in recommending ads in offline interaction scenario and develop methods to overcome these challenges. Finally, the paper presents a method to recommend ads using fashion based features with the help of computer vision and demonstrates its working.

General Terms

Personalized Advertising, Computer Vision

Keywords

Deep Learning, Power Efficient Machine Learning, Computer Vision, Personalized Advertising, Computer Vision in Embedded Systems

1. INTRODUCTION

All businesses exist to make profit. Advertisements are often a means to attract consumers towards the business. Thus advertising plays an imperative role for both manufacturers and consumers. Typically businesses use service of Ad agencies to promote their products and services. With the widespread of internet, advertising has moved online as well. Major tech companies such as Google, Instagram, Facebook provide means to promote a business online. These companies typically promote ads on commonly visited places such as popular websites, mobile apps, games, TV shows in the smartphones, browsers, TVs or other personal electronic appliances.

1.1 Offline vs Online Advertising

Online advertising can be defined as ”the act of recommending ads to users based on traceable features which rely on use of one or more personal electronic devices typically connected to and traceable via the internet”. Online advertising is a cost efficient solution for most businesses today. For example, whenever a new CPU / GPU / computer hardware is launched, it’s easy to find out a group of users on social media who have previously browsed technical content related to computer hardware.
However, advertising in some specific domains such as fashion relies on the taste of the user rather than their browsing history. For example, fashion advertisements in places such as malls, airports etc. relies highly on the assumption about the target audience visiting the mall. Typically, a large amount of money is spent in determining the ads suitable for placements at these types of places. Even then, the determined ad may not be optimal and consumers may often see ads not relevant to them. The paper will classify such advertising under ”offline” advertising.

1.2 Personalized Advertisements

Advertisers can deliver personalized advertisements to the users by observing specific traits in the consumer. In online advertising scenario, these traits are derived from browser history, cache, device specification, location etc. These types of personalized advertisements have shown to be very effective [9]. It is observed that about 40% of the consumers end up clicking on ads that are personalized for them. (Depending on the industry, the click rate or ”in-target” rate varies) This confirms that personalized advertisements are indeed more effective in capturing the correct audience.

1.3 Proposed Solution

Learning from personalized advertisements, the paper proposes a system to enhance the offline advertising. The key principle behind the personalized advertising is to ”perceive” the user in order to show relevant ads. In terms of online advertising, the act of ”perceiving” is based on user’s browser history, location, age etc. However, in terms of ”offline” advertising (as defined above), the options to ”perceive” the users are very sparse. Typically mall management or airport authorities decide what ads to show on the electronic devices mentioned previously. If this decision is not audience centered, the advertiser loses out on most of the potential consumers.

To overcome these shortcomings, the proposed system provides a means of ”perceiving” the consumer to offline advertisers in some specific situations. The proposed system uses enhances in computer vision as primary means of ”perceiving” the user. Following describes the rough workflow of proposed solution:

—Capture the image of the consumer
—Extract a set of pre-decided features (such as clothing style, approximate age and gender etc.)
—Feed these features to a recommender system to get a set of relevant ads
—Display the advertisements in the ”most-relevant” to ”least-relevant” fashion.

1.4 Scope

Although the proposed system cannot enhance *all* means of offline advertising, the paper demonstrates that the proposed system can prove to be useful in scenarios where the user interacts with an electronic device for about 1-2 minutes. These include self check-out stations at airports, arcade games at malls etc.

2. CHALLENGES

When designing such a system, following challenges must be faced:

2.1 Data Security and Privacy

The proposed system involves a camera to capture images of the user. In most countries, doing so would need a permission from user to *store* or *send* this image over the network. Therefore, in the proposed system all the computation needs to be done on-device. In fact, the stored stream of image should never even leave memory (i.e. should not be stored locally on the file system either). In other words, the system must entirely run within RAM.

2.2 Compute Limitations

The proposed system uses latest advents in deep learning and computer vision to extract fashion related attributes given image of the user. Typically these models need a lot of compute to work real time. This means without any optimizations, the targeted businesses would need to own, and more importantly maintain expensive computers (including critical components such as expensive GPU) at their ends. In general, this is a not a great design choice and hence all the deep learning models must be optimized to run on low compute devices as much as possible.

2.3 Easy to Operate

As discussed above, the targeted businesses would not prefer to have complicated compute machinery along with their devices. For example, it would be really awkward for a self check-in booth at the airport to have a 500$ GPU to perform real-time calculation. Other than cost, it is also difficult to maintain such system. Therefore, the proposed system should be robust and easy to operate.

2.4 Fair Machine Learning Models

Many machine learning models show a tendency to overfit on certain type of attribute. This leads to the model being unfair to people of certain races or gender. A thorough analysis should be done before deploying any machine learning model to make sure that the proposed system is fair to all.

3. RELATED WORK

3.1 Fashion Detection

Fashion detection is the heart of the proposed system. This is basically a multi-class classification problem where each class denotes the specific fashion attribute. A variety of deep learning classifier architectures (such as [8], [13], [10]) are available for classification. In the proposed system, authors tried various architectures to pick the best one for the problem.

3.2 Data

Data from single source can be implicitly or explicitly. It is suggested to use data from multiple datasets in hopes to cancel out bias from single data source. In particular, the authors reviewed pedestrian attribute datasets [14] and selected a subset of features from multiple datasets for demonstration later. A detailed overview of the pedestrian attribute datasets is available in Table 1 of [14]. Paper omits detailed comparison in favor of not repeating the work.

3.3 Model Compression and Optimizations

The proposed system plans to leverage deep learning, specifically convolutional neural networks [11] to extract features from the captured image. These models tend to be very heavy to compute efficiently on mobile devices. Several approaches have been thought of to compress model size and reduce model parameters. Table 1 compares main approaches for the same. In practice, authors found out that performing distillation first and then performing other optimizations such as parameter pruning/sharing, weight quantization or low rank factorization was most effective.

It is also incredibly expensive to run even compressed and optimized neural network on every single frame captured by the camera. Instead, pose detector methods such as OpenPose [7] or Lightweight OpenPose [12] can be used to detect if a person is actually walking towards the camera or just passing by. The ability of OpenPose to give scaled inference allows the system to approximate the distance of the person from camera. This enables the system to approximate the direction and distance of the person w.r.t camera. Note that this is not the recommended way to correctly predict person’s position w.r.t camera, but it is fast enough for the application.

4. COMPUTE PLATFORMS

Selecting a proper compute platform for this type of system is crucial and also a difficult question to answer. Paper reviews some of the most popular low power single board compute platforms easily available in the retail market. The results of this comparison is available in table 2 (The column ”OSS Support” refers to ”Open Source Software Support”) Despite being a specially designed board with DSP (Digital Signal Processing) cores, it is difficult to write custom neural network software for the BeagleBone AI board due to lack of any open source library to effectively communicate with it.

5. FORMAL SYSTEM DESCRIPTION

The proposed system is divided into 5 modules for better understanding and development. The system can be readily integrated with an existing kiosk or targeted ad device as shown in the figure 2

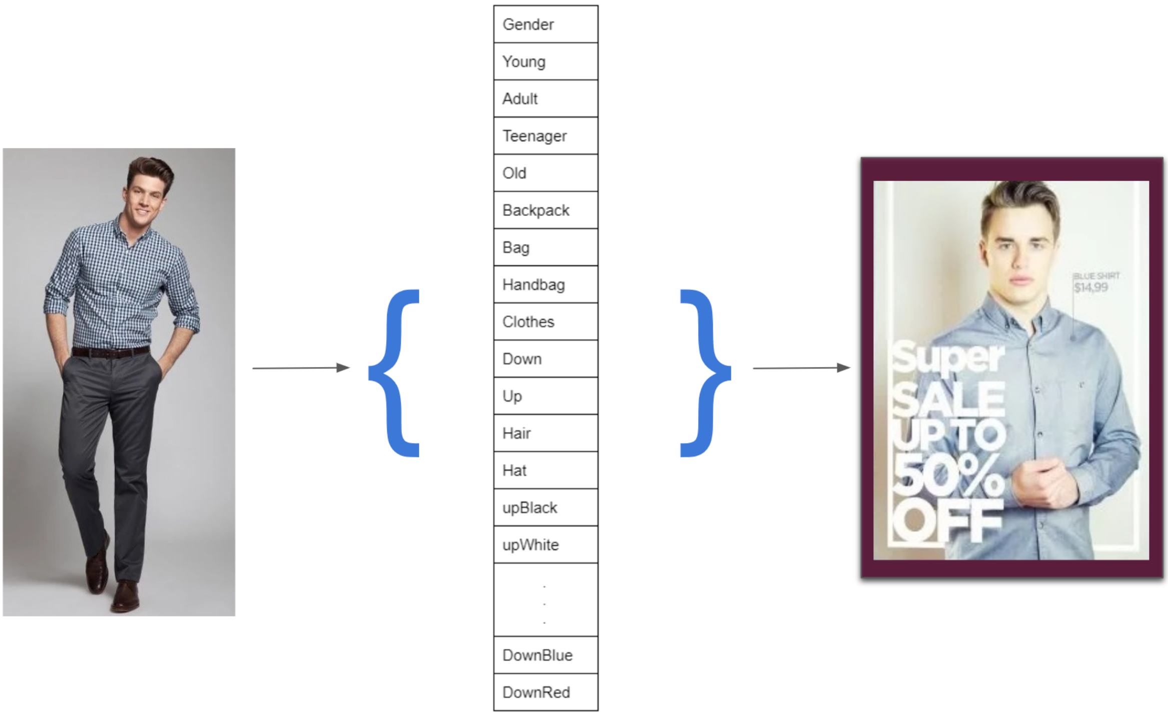


Fig. 1: High level working of the proposed system

Approach	Description	Applications	Details
Parameter pruning and sharing	Reduce parameters which are less senstitive towards performance of the model	Convolutional layers and Fully connected layers	Robust to multiple different settings, achieves decent performance, can be applied on pretrained models or during training phase itself
Low-rank factorization	Using tensor decomposition to figure out informative parameters	Convolutional layers and Fully connected layers	A stadardized pipeline that is easy to implement. Supports both pretrained models or training from scratch
Transferred / Compact Convolutional Filters	Designing special structural convolutional filters to save parameters	Convolutional layers (only)	Applications using ”Fully Convolutional Layers” can be benefitted. Only allows training from scratch
Knowledge distillation	Learning a compact neural network that learns to mimic bigger, complex network by minimizing (typically ell_2) distance between outputs of both networks	Convolutional layers and Fully connected layers	Model performance is highly sensitive towards application and network architecture. Only supports training from scratch

Table 1. : Summary of various approaches used for model compression and acceleration

5.1 Camera Module

Responsibility of this module is to capture a fixed number of frames (say n) frames using the camera sensor and store them at predefined location in memory. These frames will be automatically overridden after m clock cycles. Values of n and m will be pre-decided.

5.2 Frame Selection Module

This modules selects an appropriate frame for passing to the feature extractor module. Metrics such as pixel signal-to-noise ratio can be used to rule out particularly noisy frames. On the remaining frames, pose estimation is performed to select the optimal frame for next module.

Single Board Computer	CPU	GPU	RAM	Operating Power	OSS Support
Raspberry Pi 4 [5]	Broadcom 2711 (4xARM Cortex A72)	Broadcom VideoCore VI	4GB	5V@3A (usb-c)	Excellent
Rock64 Media Board [4]	Rockchip RK3328 (4xARM Cortex-A53)	Mali 450 MP2	4GB	5V@3A (barrel connector)	Good
PINE A-64 LTS [6]	Allwinner A64 (4xARM Cortex-A53)	Mali 400 MP2	2GB	5V@3A (barrel connector)	Good
BeagleBone AI [2]	Ti Sitara AM5729(2x ARM Cortex-A15)	PowerVR SGX544	1GB	5V@3A (usb-c)	Limited
Odroid C2 [3]	Amlogic S905 (4xARM Cortex-A53)	Mali 450 MP2	2GB	5V@3A (usb-c)	Limited

Table 2. : Overview of popular embedded single board computers

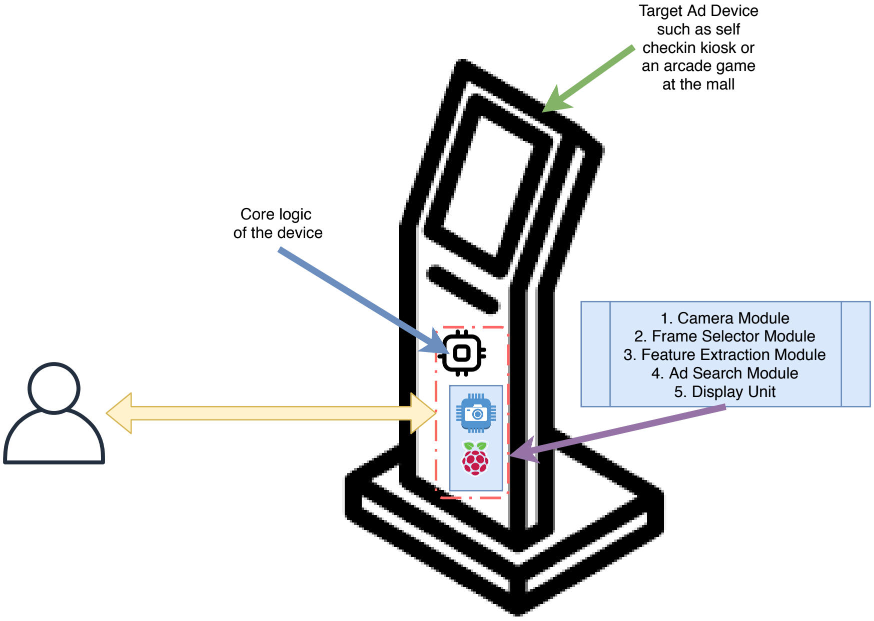


Fig. 2: High level working of the proposed system

5.3 Feature Extractor Module

This module will be a convolutional neural network which outputs an N dimensional feature vector as its logits. In the proposed system, logits consist of different types of variables as opposed to single type of variable for most image classification tasks (image classification tasks involve predicting class probabilities, which are all numeric variables (ranging from 0-1) and thus traditional loss functions such as cross entropy loss can be used to train them end-to-end). Since logits in the proposed system may include numerical variables such as age (can range from 5-∞) along with categorical variables such as `carrying_bag`, the optimization problem is slightly different and difficult.

5.4 Ad Recommendation Module

The role of this module is to present η relevant ads given the N dimensional feature vector. This primarily relies on learning the mapping between feature vector and advertisements. The easiest way of doing this would be to statically link all the possible feature vector combinations to a list of ads. However, this easy approach is very tedious and time consuming. A realistic implementation can start with the static linking algorithm. With the increase in scale, this mapping can be improved by relevance feedback algorithms such Rocchio algorithm.

5.5 Inter-Module Communication and Data Flow

Shown as a diagram in figure 3

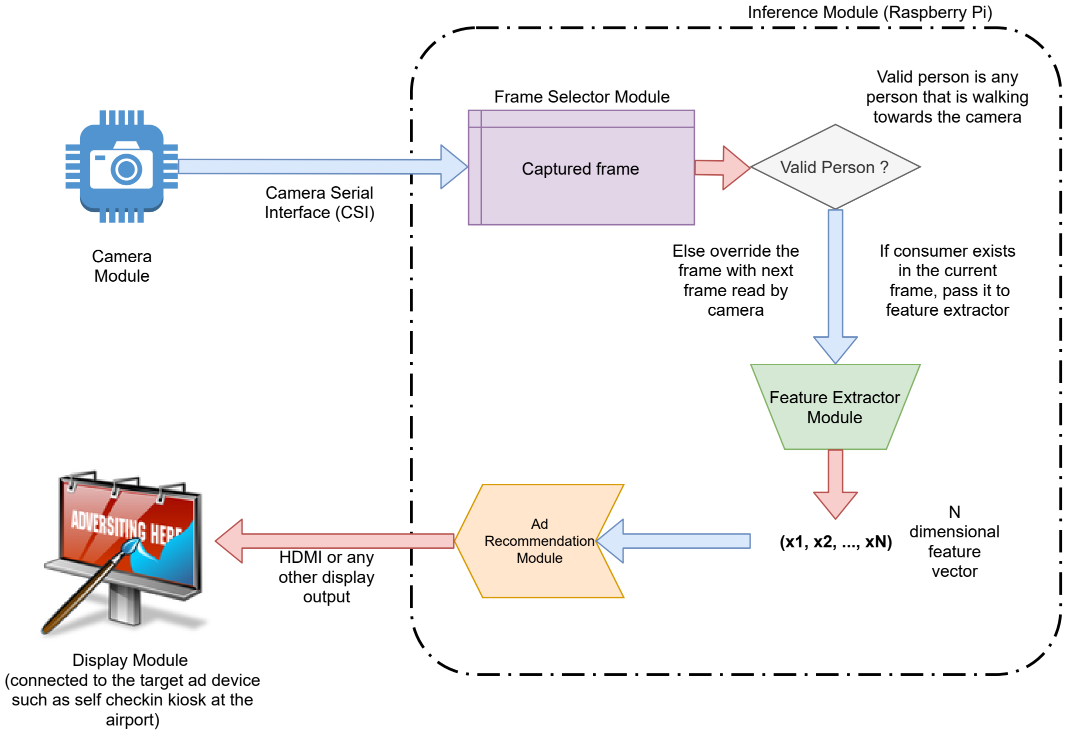

Inference Module (Raspberry Pi)
Camera Module
Camera Serial Interface (CSI)
Frame Selector Module
Captured frame
Valid Person ?
Valid person is any person that is walking towards the camera
Else override the frame with next frame read by camera
If consumer exists in the current frame, pass it to feature extractor
Feature Extractor Module
(x1, x2, ..., xN)
N dimensional feature vector
Ad Recommendation Module
HDMI or any other display output
ADVERSITING HERE
Display Module (connected to the target ad device such as self checkin kiosk at the airport)

Fig. 3: Communication Interfaces and Data Flow Diagram

6. EXPERIMENTATION

Authors use a Raspberry Pi 4B with 4GB RAM to conduct all the experiments. A ResNet50 [8] is trained on the collected data and then knowledge distillation is performed to get distilled/student model. The student model architecture used is MobileNet V2 [13]. The Ad Recommendation system currently has few hard-coded values and then some changes based on the manual feedback given. OpenPose Light [12] is used for detecting poses. In addition to these following environmental optimizations were done:

—Using Ubuntu 20.04 64 bit server (no GUI)

—Using C++ instead of Python to rule out interpreter delays

—Using ARM NEON instruction set (SIMD) [1] for better performance in deep learning operations

—Preloading and memory pinning for faster access to models in the memory

To use the neural network models in C++, `armnn` compute library with ONNX model spec was used. The video demo of the proposed system in action is available at `https://youtu.be/WhKUI_4dASw`. Final selection of fashion based features were:

(1) `carrying_bag` (refers to cross/shoulder bag) (binary attribute)
(2) `carrying_backpack` (binary attribute)
(3) `carrying_handbag` (binary attribute)
(4) `lower_body_cloth` (categorical attribute)
(5) `length_lower_body_cloth` (categorical attribute)
(6) `wearing_hat` (binary attribute)
(7) `wearing_glasses` (binary attribute)
(8) `cloth_pattern` (categorical attribute)
(9) `cloth_color` (categorical attribute)
(10) `cloth_type` (categorical attribute)

Additional attributes such as `age`, `gender` and `hair_length` (all categorical) which are not explicitly related to fashion were also inferred.
Overall, the system does following things:

—Detect if a person is walking by - Try to predict pose of the person using a pre-trained neural net (OpenPose Light [12]) and see if the person stays in frame for more than 2.5s (this timeout can be customized). The system saves the frame where the area of the predicted pose is maximum. (Figure 4a shows output after step 1)

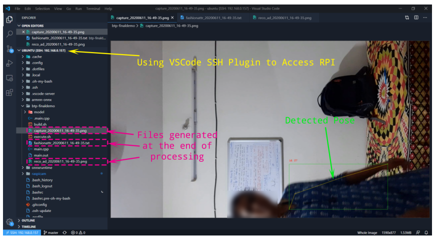
—Extract fashion attributes from the saved frame - Run the fashion classifier network on the frame saved in previous step. (Figure 4b shows output after step 2)

—Recommend ads - Pass the fashion attribute vector to the recommendation system to get recommended ads. Currently the system fetches top 3 ads. (Figure 4c shows output after final step)

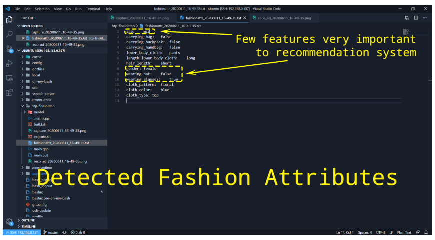
The recommendation system returns top-η ads for the corresponding feature vector. The system outputs these ads concatenated to each other for the sake of experimentation.

7. DISCUSSION AND CONCLUSION

The paper analyses and defines challenges in offline personalized advertisements. The paper also proposes a novel method for offline personalized advertisements with the help of computer vision and machine learning. Authors successfully demonstrate working

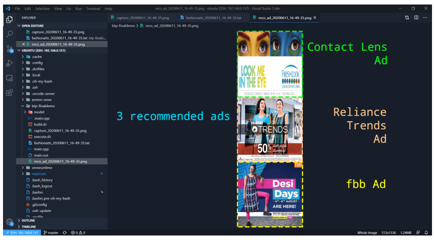

Using VSCode SSH Plugin to Access RPI
Files generated at the end of processing
Detected Person

(a) Demo: Step 1 output


Few features very important to recommendation system
Detected Fashion Attributes

```
age:    mid
carrying_bag:   false
carrying_backpack:  false
carrying_handbag:   false
lower_body_cloth:   pants
length_lower_body_cloth:    long
hair_length:    short
gender: female
wearing_hat:    false
wearing_glasses:    true
cloth_pattern:  floral
cloth_color:    blue
cloth_type: top
```

(b) Demo: Step 2 output



(c) Demo: Step 3 output

of the proposed system through experimentation. Mobile Ad developers such as Google AdSense or Facebook Ads can also use the presented approach to show ads even when the user is offline on mobile phones. While the paper successfully demonstrate the working of the proposed system, following things can be worked upon as a part of future work:

—Collecting more data with diverse fashion attributes - This will allow the fashion feature extractor network to be more robust to real world data.

—Incorporating emotion recognition as a traceable feature - Emotion recognition will help deploying offline ads when the user is extremely close to the camera (e.g. smartphone camera).

—Handling multiple users case - There could be multiple users waiting in line to use the targeted ad device. A sophisticated approach for handling this case can also be though of.

8. REFERENCES

[1] ARM NEON instruction set. http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dht0002a/ch01s03s03.html. Accessed: 2020-06-13.

[2] BeagleBone AI. https://beagleboard.org/ai. Accessed: 2020-06-13.

[3] Odroid C2. https://wiki.odroid.com/odroid-c2/odroid-c2. Accessed: 2020-06-13.

[4] Pine A64-LTS. https://www.pine64.org/devices/single-board-computers/pine-a64-lts/. Accessed: 2020-06-13.

[5] Raspberry Pi 4. https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/. Accessed: 2020-06-13.

[6] Rock64. https://www.pine64.org/devices/single-board-computers/rock64/. Accessed: 2020-06-13.

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[9] Khalid Saleh. Effectiveness of online advertising statistics and trends. [Online; Accessed 14-11-2019].

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436 EP –, May 2015.

[12] Daniil Osokin. Real-time 2d multi-person pose estimation on CPU: lightweight openpose. *CoRR*, abs/1811.12004, 2018.

[13] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv e-prints*, page arXiv:1801.04381, January 2018.

[14] Rui Yang Bin Luo Jin Tang Xiao Wang, Shaofei Zheng. Pedestrian attribute recognition: A survey. *arXiv preprint arXiv:1901.07474*, 2019.