XCloud: Design and Implementation of AI Cloud Platform with RESTful API Service

Lu Xu
Yating Wang
xulu0620@gmail.com
yt.one93@gmail.com

ABSTRACT

In recent years, artificial intelligence (AI) has aroused much attention among both industrial and academic areas. However, building and maintaining efficient AI systems are quite difficult for many small business companies and researchers if they are not familiar with machine learning and AI. In this paper, we first evaluate the difficulties and challenges in building AI systems. Then an cloud platform termed *XCloud*, which provides several common AI services in form of RESTful APIs, is constructed. Technical details are discussed in Section 2. This project is released as open-source software and can be easily accessed for late research. Code is available at https://github.com/lucasxlu/XCloud.git.

KEYWORDS

deep learning, cloud computing, computer vision, machine learning, artificial intelligence

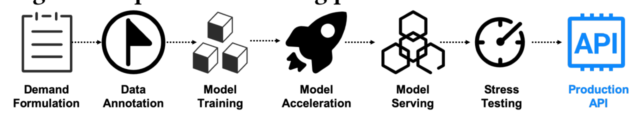
1 INTRODUCTION

Recent years have witnessed many breakthroughs in AI [5, 11, 17], especially computer vision [10], speech recognition [1] and natural language processing [7]. Deep learning models have surpassed human on many fields, such as image recognition [4] and skin cancer diagnosis [3]. Face recognition has been widely used among smart phones (such as iPhone X FaceID [1]) and security entrance. Recommendation system (such as Alibaba, Amazon and ByteDance) helps people easily find information they want. Visual search system allows us to easily get products by just taking a picture with cellphone [25, 26].

However, building an effective AI system is quite challenging [16]. Firstly, the developers should collect, clean and annotate raw data to ensure a satisfactory performance, which is quite time-consuming and takes lots of money and energy. Secondly, experts in machine learning should formulate the problems and develop corresponding computational models. Thirdly, computer programmars should train models, fine-tune hyper-parameters, and develop SDK or API for later usage. Bad case analysis is also required if the performance of baseline model is far from satifaction. Last but not least, the above procedure should be iterated again and again to meet the rapid change of requirements (see Figure 1). The whole development procedure may fail if any step mentioned above fails.

[1]https://support.apple.com/en-us/HT208109

Figure 1: Pipeline of building production-level AI service



Facing so many difficulties, cloud services (such as Amazon Web Service (AWS) [2], Google Cloud [3], AliYun [4] and Baidu Yun [5]) are getting increasingly popular among market. Nevertheless, these platforms are developed for commercial production. Researchers only have limited access to existing APIs, and cannot know the inner design architecture of the systems. So it is difficult for researchers to bridge the gap between research models and production applications.

Aiming at solving problems mentioned above. In this paper, we construct an AI cloud platform termed *EXtensive Cloud (XCloud)* with common recognition abilities for both research and production fields. *XCloud* is freely accessible and open-sourced on github [6] to help researchers build production application with their proposed models.

2 XCLOUD

In this section, we will give a detailed description about the design and implementation of *XCloud*. *XCloud* is implemented based on PyTorch [15] and Django [7]. The development of machine learning models are derived from published models [5, 6, 22–24], which is beyond the scope of this paper. The architecture of *XCloud* is shown in Figure 2. Users can upload image and trigger relevant JavaScript code, the controller of *XCloud* receive HTTP request and call corresponding recognition APIs with the uploaded image as input. Then *XCloud* will return recognition results in form of JSON. By leveraging RESTful APIs, the developers can easily integrate existing AI services into any type of terminals (such as PC web, android/iOS APPs and WeChat mini program). The overall framework of *XCloud* is shown in Figure 3.

2.1 Services

XCloud is composed of 4 modules, namely, computer vision (CV), data mining (DM) and research (R). We will briefly introduce the following services by module.

[2]https://aws.amazon.com/
[3]https://cloud.google.com/
[4]https://www.aliyun.com/
[5]https://cloud.baidu.com/
[6]https://github.com/lucasxlu/XCloud.git
[7]https://www.djangoproject.com

Figure 2: Architecture of XCloud

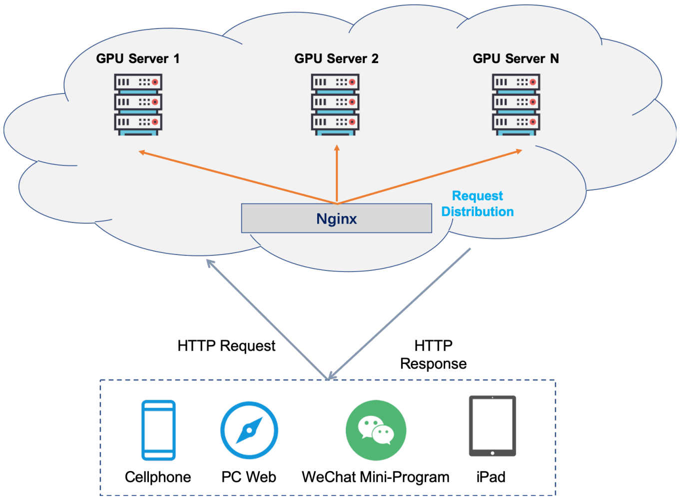
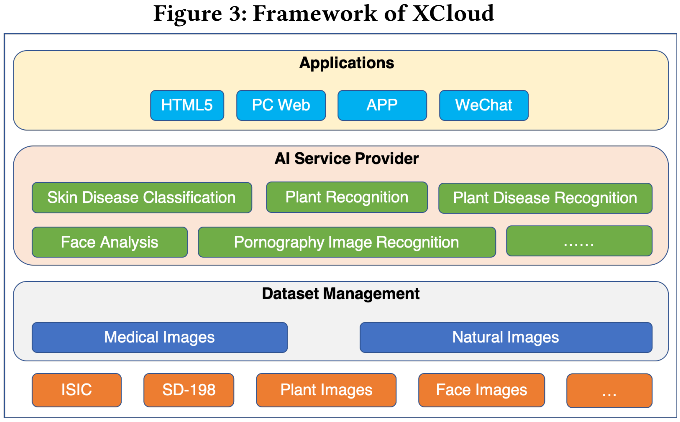


Figure 3: Framework of XCloud



2.1.1 *Computer Vision.*

In CV module, we implement and train serveral models to solve the following common vision problems.

- **Plants recognition** is popular among plant enthusiasts and botanists. It can be treated as a fine-grained visual classification problem, since a bunch of samples of different categories have quite similar appearance. We train ResNet18 [5] to recognize over 998 plants.
- **Plant disease recognition** can provide efficient and effective tools in intelligent agriculture. Farmers can know disease category and take relevant measures to avoid huge loss. ResNet50 [5] is trained to recognize over 60 plant diseases.
- **Face analysis** model can predict serveral facial attributes from a given portrait image. We take HMTNet [22] as computational backbone model. HMTNet is a multi-task deep model with fully convolutional architecture, which can predict facial beauty score, gender and race simultaneously from a unique model. Details can be found from [22].
- **Food recognition** is popular among health-diet keepers and is widely used in *New Ratailing* fields. DenseNet169 [6] is adopted to train food recognition model.
- **Skin lesion analysis** gains increased attention in medical AI areas. We train DenseNet121 [6] to recognize 198 common skin diseases.
- **Pornography image recognition** models provide helpful tools to filter sensitive images on Internet. We also integrate this feature into *XCloud*. We train DenseNet121 [6] to recognize pornography images.
- **Garbage Classification** has been a hot topic in China recently [8], it is an environment-friendly behavior. However, the majority of the people cannot tell different garbage apart. By leveraging computer vision and image recognition technology, we can easily classify diverse garbage. The dataset is collected from HUAWEI Cloud [9]. We split 20% of the images as test set, and the remaining as training set. We train ResNet152 [5] with 90.12% accuracy on this dataset.
- **Insect Pet Recognition** plays a vital part in intelligent agriculture, we train DenseNet121 [6] on IP102 dataset [21] with 61.06% accuracy, which is better than Wu et al. [21] with an improvement of 10.6%.

2.1.2 *Data Mining.*

In data mining module, we provide useful toolkit [23] related to an emerging research topic–**online knowledge quality evaluation** (like Zhihu Live [10]). This API will automatically calculate Zhihu Live's score within a range of 0 to 5, which can provide useful information for customers.

2.1.3 *Research.*

In this module, we provide the source code for training and test machine learning models mentioned above. Researchers can use the code provided to train their own models. Furthermore, we also reimplement several models (such as image quality assessment [2, 8, 9, 19], facial beauty analysis [22, 24], image retrieval [14, 20], etc.) in computer vision, which makes it easy for users to integrate these features into XCloud APIs.

2.2 Performance Metric

The performance of the above models are listed in Table 1. We adopt *accuracy* as the performance metric to evaluate classification services (such as plant recognition, plant disease recognition, food recognition, skin lesion analysis and pornography image recognition), and *Pearson Correlation (PC)* is utilized as the metric in facial beauty prediction task. Mean Absolute Error (MAE) is adopted as the metric in ZhihuLive quality evaluation task.

$$PC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{2}$$

where x_i and y_i represent predicted score and groundtruth score, respectively. n denotes the number of data samples. $\bar{x}$ and $\bar{y}$ stand for the mean of x and y, respectively. A larger PC value represents better performance of the computational model.

---

[8] http://www.xinhuanet.com/english/2019-07/03/c_138195992.htm
[9] https://developer.huaweicloud.com/competition/competitions/1000007620/introduction
[10] https://www.zhihu.com/lives/

Table 1: Performance of Computational Models on Relevant Datasets

Service	Model	Dataset	Performance	Result
Plant Recognition	ResNet18 [5]	FGVC5 Flowers [11]	Acc=0.8909	Plant category and confidence
Plant Disease Recognition	ResNet50 [5]	PDD2018 Challenge [12]	Acc=0.8700	Plant disease category and confidence
Face Analysis	HMTNet [22]	SCUT-FBP5500 [13]	PC=0.8783	Facial beauty score within $[1, 5]$
Food Recognition	DenseNet161 [6]	iFood [13]	Acc=0.6689	Food category and confidence
Garbage Classification	ResNet152 [5]	HUAWEI Cloud	Acc=0.9012	Garbage category and confidence
Insect Pet Recognition	DenseNet121 [6]	IP102 [21]	Acc=0.6106	Insect pet category and confidence
Skin Disease Recognition	DenseNet121 [6]	SD198 [18]	Acc=0.6455	Skin disease category and confidence
Porn Image Recognition	DenseNet121 [6]	nsfw_data_scraper [14]	Acc=0.9313	Image category and confidence
Zhihu Live Rating	MTNet [23]	ZhihuLiveDB [23]	MAE=0.2250	Zhihu Live score within $[0, 5]$

2.3 Design of RESTful API

Encapsulating RESTful APIs is regarded as standard in building cloud platform. With RESTful APIs, related services can be easily integrated into terminal devices such as PC web, WeChat mini program, android/iOS APPs, and HTML5, without considering compatibility problems. The RESTful APIs provided are listed in Table 2.

2.4 Backend Support

The backend of *XCloud* is developed based on Django [15]. We follow the *MVC* [12] design pattern which represents that the view, controller and model are separately developed and can be easily extended in later development work. In order to record user information produced on *XCloud*, we construct 2 relational tables in MySQL which is listed in Table 3 and Table 4, to store relevant information.

In addition, we also provide simple and easy-to-use script to convert original PyTorch models to TensorRT [16] models for faster inference. TensorRT is a platform for high-performance deep learning inference. It includes a deep learning inference optimizer and runtime that delivers low latency and high-throughput for deep learning inference applications. With TensorRT, we are able to run DenseNet169 [6] with 97.63 FPS on two 2080TI GPUs, which is significantly faster than its counterpart PyTorch naive inference engine (29.45 FPS).

2.5 Extensibility

As shown by the name of XCloud (EXtensive Cloud), it is also quite easy to integrate new abilities. Apart from using existing AI technology provided by *XCloud*, developers can also easily build their own AI applications by referring to the model training code contained in research module [17]. Hence, the developers only need to prepare and clean dataset. After training your own models, your AI interface is automatically integrated into *XCloud* by just writing a new controller class and adding a new Django view.

[15]https://www.djangoproject.com/

[16]https://developer.nvidia.com/tensorrt

[17]https://github.com/lucasxlu/XCloud/tree/master/research

2.6 API Stress Testing

The performance and stability play key roles in production-level service. In order to ensure the stability of *XCloud*, Nginx [18] is adopted for load balancing. In addition, we use JMeter [19] to test all APIs provided by *XCloud*. The results of stress testing can be found in Table 5.

From Table 5 we can conclude that the performance and stability of *XCloud* are quite satisfactory under current software and hardware condition. We believe the performance could be heavily improved if stronger hardware is provided. The test environment with 2080TI GPUs and Intel XEON CPU is enough to support 20 QPS (query per second). By deploying *XCloud* on your machine and running server, you will get the homepage as Figure 4.

Figure 4: Homepage of *XCloud*

3 CONCLUSION AND FUTURE WORK

In this paper, we construct an AI cloud platform with high performance and stability which provides common AI service in form of RESTful API, to ease the development of AI projects. In our future work, we will integrate more service into *XCloud* and develop better models with advanced performance.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[18]http://nginx.org/

[19]https://jmeter.apache.org/

Table 2: Definition of RESTful API

API	Description	HTTP Methods	Param
cv/mcloud/skin	skin disease recognition	POST	imgraw/imgurl
cv/fbp	facial beauty prediction	POST	imgraw/imgurl
cv/nsfw	pornography image recognition	POST	imgraw/imgurl
cv/pdr	plant disease recognition	POST	imgraw/imgurl
cv/food	food recognition	POST	imgraw/imgurl
cv/plant	plant recognition	POST	imgraw/imgurl
cv/facesearch	face retrieval	POST	imgraw/imgurl
dm/zhihuliveeval	Zhihu Live rating	GET	Zhihu Live ID

Table 3: API calling details table. The primary key is decorated with underline.

Attribute	Type	Length	Is Null?
<u>username</u>	varchar	16	False
api_name	varchar	20	False
api_elapse	float	10	False
api_call_datetime	datetime	-	False
terminal_type	int	3	False
img_path	varchar	100	False

Table 4: User information table. The primary key is decorated with underline.

Attribute	Type	Length	Is Null?
<u>username</u>	varchar	16	False
register_datetime	datetime	-	False
register_type	int	11	False
user_organization	varchar	100	False
email	varchar	50	False
userkey	varchar	20	False
password	varchar	12	False

Table 5: Stress Testing Results on NVIDIA 2080TI GPU

API	AVG_LATENCY (ms)	P99 (ms)	ERROR
cv/mcloud/skin	16	20	0
cv/fbp	25	36	0
cv/nsfw	16	21	0
cv/pdr	16	23	0
cv/food	17	23	0
cv/plant	18	25	0
dm/zhihuliveeval	5	8	0

[2] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. A deep neural network for image quality assessment. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3773–3777. IEEE, 2016.

[3] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[7] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.

[8] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1733–1740, 2014.

[9] Le Kang, Peng Ye, Yi Li, and David Doermann. Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. In *2015 IEEE international conference on image processing (ICIP)*, pages 2791–2795. IEEE, 2015.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[12] Avraham Leff and James T Rayfield. Web-application development using the model/view/controller design pattern. In *Proceedings fifth ieee international enterprise distributed object computing conference*, pages 118–127. IEEE, 2001.

[13] Lingyu Liang, Luojun Lin, Lianwen Jin, Duorui Xie, and Mengru Li. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1598–1603. IEEE, 2018.

[14] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[16] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.

[17] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[18] Xiaoxiao Sun, Jufeng Yang, Ming Sun, and Kai Wang. A benchmark for automatic visual classification of clinical skin disease images. In *European Conference on Computer Vision*, pages 206–222. Springer, 2016.

[19] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018.

[20] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[21] Xiaoping Wu, Chi Zhan, Yu-Kun Lai, Ming-Ming Cheng, and Jufeng Yang. Ip102: A large-scale benchmark dataset for insect pest recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8787–8796, 2019.

[22] Lu Xu, Heng Fan, and Jinhai Xiang. Hierarchical multi-task network for race, gender and facial attractiveness recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3861–3865. IEEE, 2019.

[23] Lu Xu, Jinhai Xiang, Yating Wang, and Fuchuan Ni. Data-driven approach for quality evaluation on knowledge sharing platform. *arXiv preprint arXiv:1903.00384*, 2019.

[24] Lu Xu, Jinhai Xiang, and Xiaohui Yuan. Crnet: Classification and regression neural network for facial beauty prediction. In *Pacific Rim Conference on Multimedia*, pages 661–671. Springer, 2018.

[25] Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. Visual search at ebay. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2101–2110. ACM, 2017.

[26] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. Visual search at alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 993–1001. ACM, 2018.