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

Lu Xu, Yating Wang

E-MAIL: 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 named *XCloud*, which provides several common AI services in form of RESTful APIs, is constructed. Technical details, feasibility analysis and performance test settings are discussed as well. 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 [4, 8], especially computer vision [7], speech recognition [1] and natural language processing [6]. Deep learning models have surpassed human on image recognition [3], skin cancer diagnosis [2] and many other fields. Face recognition has been widely used among smart phones and security entrance.

However, building an effective AI system is quite challenging. Firstly, the developers should collect, clean and annotate raw data to ensure a satisfactory performance. Secondly, experts in machine learning should formulate the problems and develop computational models. Thirdly, computer programmars train models and develop SDK 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. The whole development may fail if any step mentioned above fails.

Facing so many difficulties, cloud services (such as Amazon Web Service (AWS) [1], Google Cloud [2], AliYun [3] and Baidu Yun [4]) are getting popular among market. Nevertheless, these platforms are only for commercial usage. Researchers only have limited access to existing APIs, and cannot know the inner design architecture of the systems.

In this paper, we construct an AI cloud platform with common recognition abilities for research. *XCloud* is free of charge and open-sourced on github, hence researchers have easy access to the platform.

2 XCloud

In this section, we will give a detailed description about the design and architecture of *XCloud*. The development of machine learning models are derived from current state-of-the-art models [4, 5] and our previous works [18, 16, 17], which are beyond the scope of this paper. The architecture of *XCloud* is shown in Figure 1. Users can upload image and trigger relevant JavaScript code, the controller of *XCloud* receive HTTP request and call recognition APIs with the uploaded image. Then *XCloud* will return recognition results in form of JSON. By leveraging RESTful APIs, the developers can easily integrate 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 2.

2.1 Services

XCloud is composed of 4 modules, namely, computer vision (CV), natural language processing (NLP), data mining (DM)

[1] `https://aws.amazon.com/`
[2] `https://cloud.google.com/`
[3] `https://www.aliyun.com/`
[4] `https://cloud.baidu.com/`

FIGURE 1. Architecture of *XCloud*

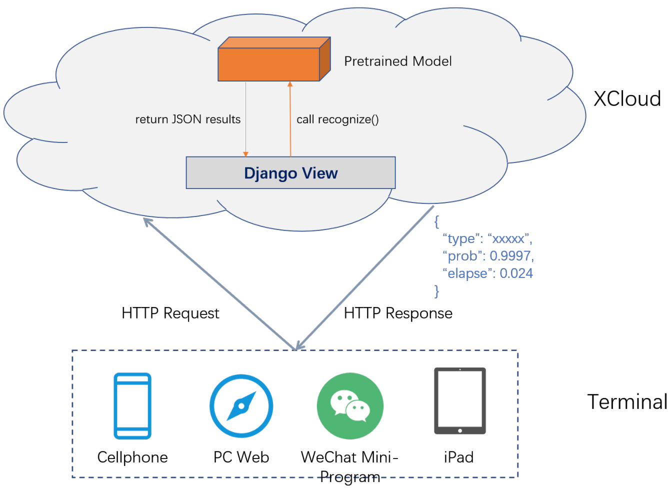
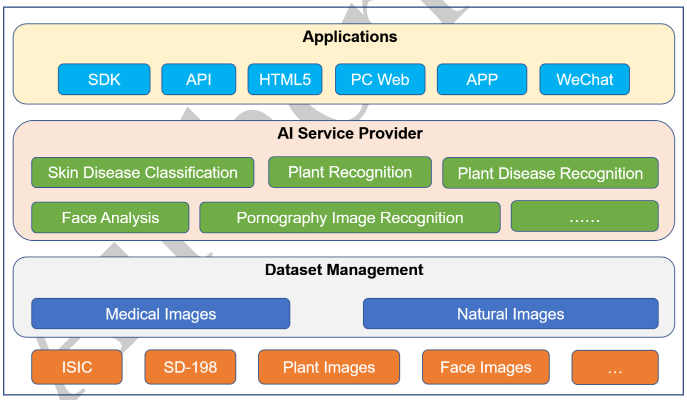


FIGURE 2. Framework of *XCloud*



and research. We will briefly introduce the following services by module.

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 [4] to recognize over 998 plants.
- **Plant disease recognition** can provide efficient and effective tools in intelligent agriculture. Farmers can know what disease plant has and take relevant measures to avoid huge loss. ResNet50 [4] is trained to recognize over 60 plant diseases.
- **Face analysis** model can predict serveral facial attributes from a given portrait image. We take HMTNet [16] 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. Details can be found from [16].
- **Food recognition** is popular among health-diet keepers and is widely used in *New Ratailing* fields. DenseNet169 [5] is adopted to train food recognition model.
- **Skin lesion analysis** gains increased attention in medical AI areas. We train DenseNet121 [5] to recognize 198 common skin diseases.
- **Pornography image recognition** models provide helpful tools to filter sensitive images in Internet. We also integrate this function into *XCloud*. We train DenseNet121 [5] to recognize pornography images.
- **Face Retrieval** is widely adopted in security entrance, we also integrate face similarity search into *XCloud*. We pretrain SphereFace [11] model to extract facial features, and store them in a binary file. Users can upload an image and *XCloud* will return Top-10 portrait images which are most similar to the uploaded image.
- **Garbage Classification** is a hot topic in China recently [5], it is an environment-friendly behavior. However, the majority of the people cannot tell different garbage apart.

[5] `http://www.xinhuanet.com/english/2019-07/03/c_138195992.htm`

By leveraging computer vision and image recognition technology, we can easily classify diverse garbage. The dataset is collected from HUAWEI Cloud [6]. We split 20% of the images as test set, and the remaining as training set. We train ResNet152 [4] with 90.12% accuracy on this dataset.

- **Insect Pet Recognition** plays a vital part in intelligent agriculture, we train DenseNet121 [5] on IP102 dataset [15] with 61.06% accuracy, which is better than Wu et al. [15] with an improvement of 10.6%.

2.1.2 NLP

Sentiment analysis and **news classification** are fundamental parts in NLP tasks. Hence we provide basic sentiment analysis based on snownlp library [7], which is mainly designed and trained for Chinese corpus. In addition, we train DAE-RF algorithm to classify news category with 82.09% accuracy. DAE-RF algorithm is a hybrid model which combines the feature learning ability of auto encoder and powerful classification ability of random forests.

Furthermore, we also integrate **hot words analysis** into *XCloud* NLP module with the help of jieba library [8]. The TOP-K hot words are generated via *TF-IDF weights* and *TextRank* [12].

Take Figure 3 for example, by visiting hot words analysis page, the users just need to copy and paste the text content into text area, and then click "calculate" button. *XCloud* will automatically calculate TOP-K hot words and visualize them in wordcloud.

2.1.3 Data Mining

In data mining module, we provide serveral data acquire interface and emerging research topic–**online knowledge quality evaluation** (like Zhihu Live [9]). This API will automatically calculate Zhihu Live's score within a range of 0 to 5, which can provide useful information for customers.

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

FIGURE 3. Experience Page on *XCloud* Hot Words Analysis

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.

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

The backend of *XCloud* is developed based on Django [15]. We follow the *MVC* [9] design pattern which represents that the

[6]https://developer.huaweicloud.com/competition/competitions/1000007620/introduction
[7]https://github.com/isnowfy/snownlp.git
[8]https://github.com/fxsjy/jieba.git
[9]https://www.zhihu.com/lives/

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

TABLE 1. Performance of Computational Models on Relevant Datasets

Service	Model	Dataset	Performance	Result
Plant Recognition	ResNet18 [4]	FGVC5 Flowers [10]	0.8909	Plant category and confidence
Plant Disease Recognition	ResNet50 [4]	PDD2018 Challenge [11]	0.8700	Plant disease category and confidence
Face Analysis	HMTNet [16]	SCUT-FBP5500 [10]	0.8783	Facial beauty score within $[1, 5]$
Food Recognition	DenseNet161 [5]	iFood [12]	0.6689	Food category and confidence
Garbage Classification	ResNet152 [4]	HUAWEI Cloud	0.9012	Garbage category and confidence
Insect Pet Recognition	DenseNet121 [5]	IP102 [15]	0.6106	Insect pet category and confidence
Skin Disease Recognition	DenseNet121 [5]	SD198 [14]	0.6455	Skin disease category and confidence
Porn Image Recognition	DenseNet121 [5]	nsfw_data_scraper [13]	0.9313	Image category and confidence
Face Retrieval	SphereFace [11]	HZAU-MS-Face	-	Top-10 most similar portrait images
Zhihu Live Rating	MTNet [17]	ZhihuLiveDB [17]	0.2250	Predicted score of Zhihu Live
News Classification	DAE-RF	Fudan Corpus [14]	0.8209	News category of the text content

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
nlp/hotwords	hot words extraction	GET	text content
nlp/sentiment	sentiment analysis	GET	text content
nlp/newsclassify	news classification	GET	text content
dm/zhihuliveeval	Zhihu Live rating	GET	Zhihu Live ID

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.

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

Attribute	Type	Length	Is Null?
username	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?
username	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

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. TensorRT-based applications perform up to 40x faster than CPU-only platforms during inference.

2.5 Extensibility

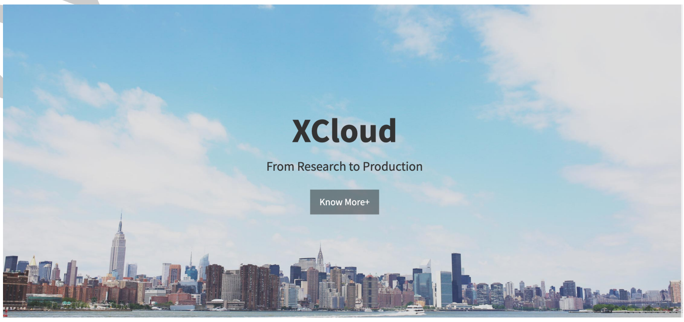
Just shown by the name of XCloud (eXtensive Cloud), it is also quite easy to integrate new functions. Apart from using existing AI technology provided by *XCloud*, developers can also easily build their own AI applications by referring to the model training code contained in Research Branch [17]. Hence,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

2.6 Testing

The performance and stability play key roles in practical usage. In order to ensure the stability of *XCloud*, we use JMeter [18] to test all APIs provided by *XCloud*. *XCloud* is implemented with PyTorch [13] and Django on an Ubuntu server equipped with NVIDIA 2080TI GPU. The testing details 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. 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

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

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

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

TABLE 5. Test Results of *XCloud* on NVIDIA 2080TI GPU

API	AVG_LOAD_TIME	AVG_CONNECT_TIME	ERROR_COUNT
cv/mcloud/skin	96	31	0/100
cv/fbp	2970	18	0/100
cv/nsfw	110	32	0/100
cv/pdr	50	15	0/100
cv/food	180	36	0/100
cv/plant	60	46	0/100
nlp/hotwords	44	107	0/100
nlp/sentiment	206	292	0/100
nlp/newsclassify	44	304	0/100
dm/zhihuliveeval	12	156	0/100

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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[14] 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.

[15] 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.

[16] Lu Xu, Heng Fan, and Jinhai Xiang. Hierarchical multi-task networks for race, gender and facial attractiveness recognition. In *The 26th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019.

[17] 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.

[18] 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.