

CVFlow: Open Workflow for Computer Vision

Xiaoze Yao
yaoxiaozhe@email.szu.edu.cn
Shenzhen University
Shenzhen, China

Yingying Chen
yy.chen@autoai.org
Shenzhen University
Shenzhen, China

Jia Wu
jia.wu@szu.edu.cn
Shenzhen University
Shenzhen, China

ABSTRACT

In recent years, the multimedia community has witnessed an emerging usage of deep learning in computer vision, such as image caption[11], object detection, instance segmentation and etc, and many companies are adopting deep learning into their commercial products. Developing and Deploying a fast and reliable deep learning system is complex and involves labour-intensive works, like image annotation[8], data and models versioning and etc. To facilitate such a process, we proposed CVFlow, which is a suite of versatile computer vision libraries and toolkits for handling computer vision tasks, including intelligent annotation toolkit, deep learning package manager, models sharing platform and computer vision serving program.

CCS CONCEPTS

• **Software and its engineering** → **Software notations and tools.**

KEYWORDS

Computer Vision, Package Manager, Deep Learning, Industrial Application

ACM Reference Format:

Xiaoze Yao, Yingying Chen, and Jia Wu. 2019. CVFlow: Open Workflow for Computer Vision. In *Nice '19: ACM International Conference on Multimedia, October 21–25, 2019, 2019, Nice, France*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

In the past few years, deep learning has played an important role in solving computer vision tasks, such as image caption, image generation, image classification and etc. Developing deep learning applications usually begins from collecting and annotating datasets, finding and optimizing neural networks to evaluating performance on test datasets, and deployments. There appeared many frameworks, such as Tensorflow[1], PyTorch[7], Chainer[10], MXNet[3] and etc.

The growing development of frameworks and libraries has greatly reduced the effort of designing, implementing and evaluating neural networks. However, the bottleneck still exists in large-scale datasets

annotation, model sharing and deployments. Creating and annotating large datasets is labour-intensive and involves significant time and costs[2], which makes it unaffordable for small companies. Besides, the main approach to sharing and redistribution of deep learning models and algorithms is redistributing the source code and weight file, which causes incompatibilities and conflicts and brings extra and prerequisites knowledge when deploying computer vision services. To reduce the time and efforts for annotating datasets, versioning, deploying and sharing models, we proposed and developed CVFlow for Open Computer Vision Workflow. It is a suite of toolkits and libraries that currently provides (1) intelligent annotation that can help annotators improve efficiency by providing suggestions (2) package manager that manages models, datasets, services and serving environment. and (3) model hub for downloading, sharing and discovering existing computer vision models.

CVFlow contains two independent but collaborative modules, which are CVPM for package management and model sharing, as well as IAE for intelligent and efficient annotation. In the following, we will describe these modules respectively.

CVPM is the abbreviation of computer vision package manager. It allows users to download, share, and install new computer vision packages. Traditional computer vision libraries often provide some high-level APIs for development such as torch vision, msvc and etc. On the contrary, CVPM provides a package management system which allows users to download computer vision packages across numerous frameworks and apply them into a Restful HTTP API[9]. By doing so, users can install and deploy computer vision models and services with few clicks or commands, without writing any code. Meanwhile, applying computer vision services into their products would be much easier since CVPM provides a similar restful API across many different libraries.

Lots of big companies have realised that one of the biggest bottlenecks in modern machine learning applications is the growing need for large, labelled datasets. Creating and annotating the datasets requires significant cost and time[2]. To facilitate such a need, we have included Integrated Annotation Environment (IAE) for intelligent visual objects annotation.

Though there are some online annotation platforms that help organisations and teams to annotate datasets effectively, they merely focus on the collaborative annotation. But, few of them pay attention to intelligent annotation. There is no reason that the annotation itself has not been transformed by artificial intelligence, as in other diverse fields. By providing various features such as intelligent annotation suggestions and automatic data augmentation, IAE helps users to annotate more effectively and more intelligently. The benefits of such features are two-folded. Firstly, by using pre-trained neural networks, users will receive suggestions before they annotate the image. If the suggestions are correct and accurate enough, users

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Nice '19, October 21–25, 2019, Nice, France

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-9999-9/18/06...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

can directly apply these suggestions to the annotation data. Otherwise, they can make minor fixes to these suggestions. Therefore the time taken on annotation will be greatly reduced by this feature. Secondly, by introducing these intelligent suggestions, users would know the accuracy of the fine-tuned model during the process of annotation. When users are annotating the datasets, IAE will fine-tune the model after a specific number of images have been annotated, making suggestions more and more accurate as the number of annotated images are growing. Benefitting from this feature, users will be able to get the desired model earlier than before by labelling fewer images because they can fine-tune the model during annotating, instead of annotating a large number of images and then training the model.

CVFlow was released on GitHub¹ in 2018, Since then the proposed workflow has been adopted by several online computer vision platforms, including training and annotating platforms.

2 ARCHITECTURE

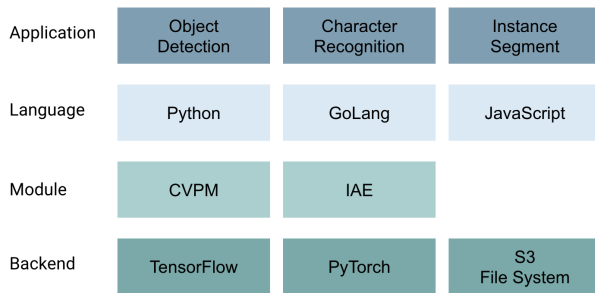


Figure 1: The Architecture of CVFlow

The architecture of CVFlow is as shown in Fig.1. Deep learning researchers, developers and companies benefit from the services of CVFlow. These services range from managing datasets and projects to annotating large-scale images, to building neural networks, to training a model using public or annotated private datasets, to managing the model versions and lifecycles, to testing and deploying deep learning models and also to publishing models to a public registry.

2.1 CVPM

CVPM provides package management for computer vision. CVPM regards an inference program as a solver which is the minimal unit of a computer vision service. Each solver is typically a Python class that implements constructor to initialize the model, and an infer function to return the predicted results. Many solvers, along with pre-trained files, configurations, and other assets, combine together to form a repository or a package which is considered to be a service for a specific computer vision task. For example, in the model hub, there is a “Face Utility” repository which is specifically implemented for face-related tasks and consists of three different

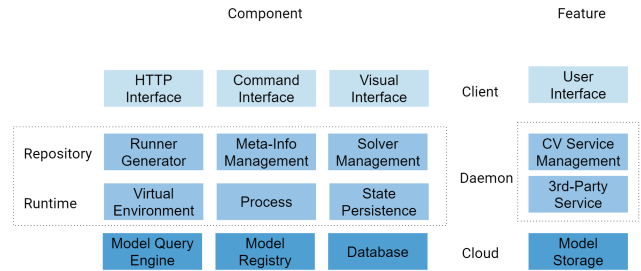


Figure 2: The Architecture of CVPM

solvers, such as face detection, face landmark recognition, and face encoding.

Inside CVPM, there are several submodules as described in the Fig.2. Users directly interact with the client module using the command line interface or web-based dashboard, to send HTTP requests to the daemon. The daemon program handles the requests and manages repositories. The daemon program has two major components which are the repository and runtime management.

The repository component is in charge of the installation, meta-information and file management of repositories. When users request to install a new package, this component will select the right protocol (GIT or HTTP protocol), fetch the meta information (including pre-trained weights, solvers, and introduction), download the required files, and then generate a runner file which indicates how to serve the solver on a specific port.

The runtime component is the core component of CVPM, which handles the state management, virtual environment, process, and environment variables of the running solvers. Every time the repository component receives users’ request to startup a solver, it will forward the request to the runtime component which handles this request. The runtime component will then try to start the solver and append the repository and solver to its persistence layer which saves the state of running repositories and solvers for further enquiries if the solver starts successfully. A typical workflow for using cvpm is shown as in Fig.3. When starting a solver the runtime component also handles the running environment, including the virtual environment and environment variables. It has built-in support for Venv which is a popular tool to create isolated Python environments, and this support allows users to keep their repositories running in isolated environments, and avoid conflicts and problems when two different repositories ask for incompatible dependencies.

Besides, CVPM also provides an open source, self-hostable cloud service that manages the model hub and registry, where developers and researchers can share their models with other users. Compared with other cloud services like Google or Azure, CVPM allows user-generated content and provides more options for handling computer vision tasks. Just like other package managers such as NPM[13] for Nodejs and apt[4] for Debian, CVPM allows users to download, install and deploy computer vision services within a few clicks.

¹<https://github.com/unarxiv/cvflow>

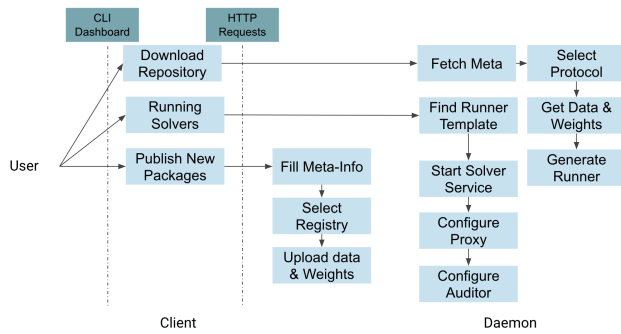


Figure 3: Typical Workflow of CVPM

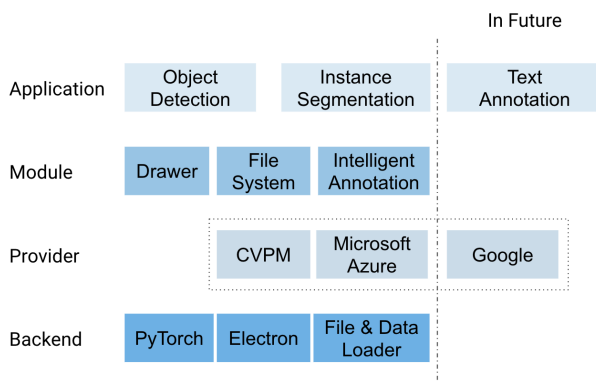


Figure 4: The Architecture of IAE

2.2 IAE

The aim of IAE is to provide an intelligent, integrated environment for annotation workflow. For now, IAE is mainly designed for computer vision annotation tasks. Since the design principle of this is low-coupling, in the future, it could support annotating text field and also be able to support different file systems easily.

The architecture of IAE is illustrated in Fig.4. It is built on top of Electron which also supports many other popular desktop applications such as VSCode, VoTT and etc. The file loader will firstly read image data into the workplace, and then the intelligent annotation will call the providers and get annotation suggestions. Afterwards, the drawer will render these suggestions and users can accept or decline these suggestions. If the suggestions are accepted, it will be turned into final annotations and will be used for further training.

3 APPLICATIONS

In this section, we present a study on the applications of CVFlow.

Large-Scale Datasets Annotation Annotation on very large datasets is currently a bottleneck for the application of machine learning, and it is a labour-intensive task which can be solved by artificial intelligence and collaborative annotation.

Transfer Learning Transfer Learning is a popular method where a model trained on one task is re-used and fine-tuned for other

tasks[6]. Inside CVFlow, we provide several pre-trained models, including fast-rcnn, single-shot box, etc., for fine-tuning. An open platform has been built to provide annotation, training service, whose training backend is provided by CVFlow, which we will discuss more in the Case Study.

Model Deployment and Sharing Compared with other model markets, we present an open-source, distributed package manager and sharing platform. Users are allowed to use GitHub (or other Git-based platforms) or our platform to host their packages, rather than limited to our platform. It also enables users to download and deploy machine learning service to bare metals instead of cloud service providers to help them reduce cost. Besides, authors of deep learning algorithms can benefit from the sharing process, if their models are adopted by large companies, which will lead the computer vision society to blossom

4 INDUSTRY AI: A CASE STUDY

IndustryAI IndustryAI, an collaborative annotation platform, has adopted CVFlow as its backend. It provides a web-based annotation platform, where a team of data annotator, reviewer, and deep learning programmer can work closely on object detection, instance segmentation or image caption tasks.

Inside IndustryAI, there is a training component which allows users to upload their annotated datasets to the training server directly from the web browser. Driven by CVPM, the training component has included several pre-defined neural networks and correspondent pre-trained weights which have been proven to be capable of general object detection[12] and instance segmentation[5] tasks. In addition, it has an inference engine which loads the fine-tuned weight and provisions an inference service. The provisioning process is driven by CVPM.

5 DISCUSSION

CVFlow provides a wide variety of features for computer vision tasks, from intelligent collaborative visual objects annotation, to datasets management, to model training and fine-tuning, to model sharing and deployments. It basically includes a whole workflow for modern computer vision services.

Features of CVFlow are provided by its modules, namely CVPM, and IAE. They cooperate closely to support the handling of computer vision tasks. As stated in the architecture of IAE, it depends on CVPM to provide pre-trained models and weights. However, the main design principle of CVFlow is its low-coupling, which means each module can be installed and worked independently. This feature offers users more options and could help them integrate any part of CVFlow to their current infrastructure, rather than being bound to a single platform.

The intended audience includes annotators who want to label datasets more accurately and efficiently, deep learning researchers who need to publish their deep learning algorithms and pre-trained weights, and companies who are trying to deploy a stable machine learning services effectively.

In future, we plan to extend the use of IAE to other fields that need annotation, such as natural language processing.

ACKNOWLEDGMENTS

The work is partially supported by the Science Foundation of Shenzhen (Grant No. JCYJ20150324141711699 and 20170504160426188). We would extend our sincere gratitude to Prof. Shiqu Yu, Mr Basker George and open source community for numerous fruitful discussions.

REFERENCES

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.
- [2] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alexander Ratner, Braden Hancock, Houman Alborzi, et al. 2018. Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. *arXiv preprint arXiv:1812.00417* (2018).
- [3] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. 2015. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274* (2015).
- [4] O Felicio de Sousa, MA de Menezes, and Thadeu JP Penna. 2009. Analysis of the package dependency on debian gnu/linux. *Journal of Computational Interdisciplinary Sciences* 1, 2 (2009), 127–133.
- [5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.
- [6] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.
- [7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.
- [8] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher RÄI. 2017. Snorkel. *Proceedings of the VLDB Endowment* 11, 3 (Nov 2017), 269–282. <https://doi.org/10.14778/3157794.3157797>
- [9] Leonard Richardson and Sam Ruby. 2007. *Restful Web Services* (first ed.). O'Reilly.
- [10] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, Vol. 5. 1–6.
- [11] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [12] Paul Viola, Michael Jones, et al. 2001. Rapid object detection using a boosted cascade of simple features. *CVPR (1)* 1 (2001), 511–518.
- [13] E. Wittern, P. Suter, and S. Rajagopalan. 2016. A Look at the Dynamics of the JavaScript Package Ecosystem. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. 351–361. <https://doi.org/10.1109/MSR.2016.043>