



# 人工智能与大数据综合实践 ——猫狗分类问题

□ 朱语、沈骏一、宋孟炫

□ 2023.07.14

# 目 录

*CONTENTS*

- 1** 选 题 背 景
- 2** 基 础 成 果
- 3** 创 新 拓 展
- 4** 总 结 展 望



选题背景

阶段性成果

创新拓展

总结与展望

## 选题背景

*Research Background*



选题背景

阶段性成果

创新拓展

总结与展望

## 1 项目的意义与目的



通过该项目，可以学习如何使用华为MindSpore框架构建一个分类模型。熟悉除了Pytorch、Tensorflow以外的模型架构。



在这个项目中，通过使用MindSpore Lite框架，可以将训练好的分类模型进行轻量化处理



通过对以上示例项目的开发，系统掌握类似基于深度学习的多目标/单目标分类项目的开发流程



选题背景

阶段性成果

创新拓展

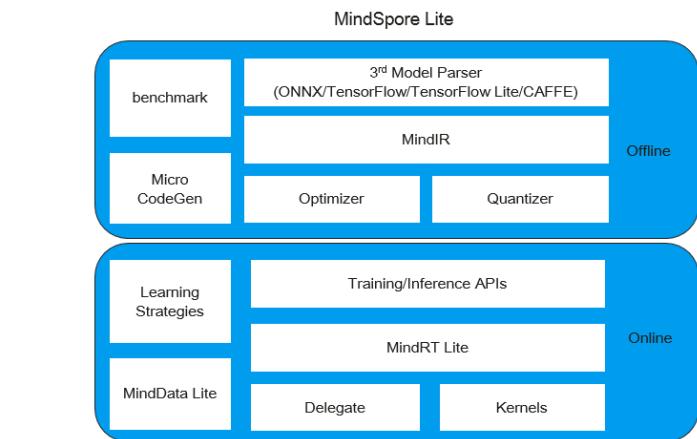
总结与展望

# 1 解决方案

本实验主要分为两部分，其中第一部分是基于MindSpore框架，通过微调训练一个猫狗分类模型，并部署到手机端；第二部分是解析APP构建代码，从而学习模型部署的执行逻辑。

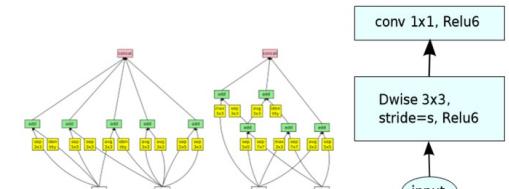
## Mindspore Lite

*MindSpore Lite*是一个极速、极智、极简的AI引擎，使能全场景智能应用，为用户提供端到端的解决方案，帮助用户使能AI能力。



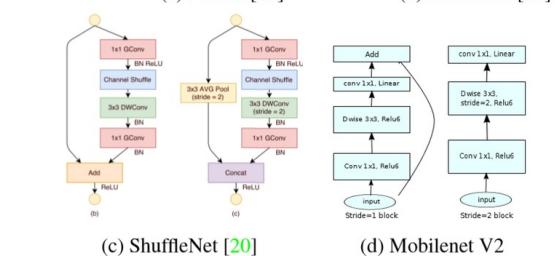
## MobileNet V2

MobileNetV2模型是专门为移动和嵌入式设备设计的网络架构，该架构能在保持类似精度的条件下显著的减少模型参数和计算量。



(a) NasNet[23]

(b) MobileNet[27]



(c) ShuffleNet [20]

(d) MnasNet V2



选题背景

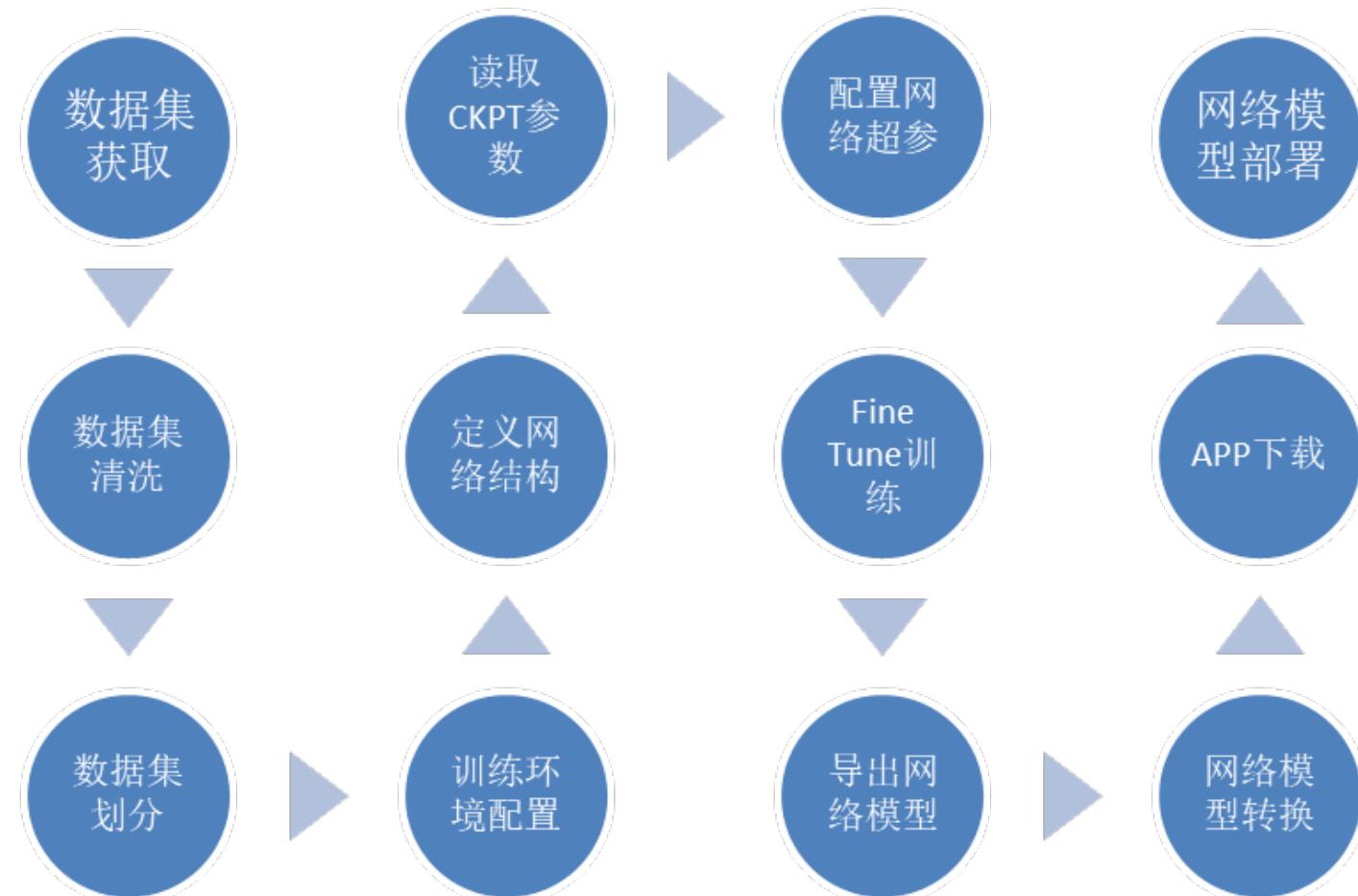
阶段性成果

创新拓展

总结与展望

## 1 拟采用的解决方案

### 项目流程图





选题背景

阶段性成果

创新拓展

总结与展望

基础成果



选题背景

阶段性成果

创新拓展

总结与展望

## 2 基础成果

### 实验平台搭建

#### 本地PC

使用anaconda新建python3.7环境，在mindspore官网参考安装命令配置环境

一、获取安装命令

版本 2.0.0 1.10.1 Nightly

硬件平台 Ascend 910 Ascend 310 GPU CUDA 10.1 GPU CUDA 11.1 CPU

操作系统 Linux-aarch64 Linux-x86\_64 Windows-x64 MacOS-aarch64 MacOS-x86\_64

编程语言 Python 3.7 Python 3.8 Python 3.9

安装方式 Pip Conda Source Docker Binary

安装命令 conda install mindspore-cpu=1.10.1 -c mindspore -c conda-forge  
# 注意参考下方安装指南，添加运行所需的环境变量配置

#### 华为云端

我们到华为云平台ModelArt上创建了实验用Notebook，使用的配置为：  
(硬件)

Ascend: 1\*Ascend910|CPU: 24核 96GB  
(软件)

tensorflow1.15+mindspore1.7.0+cann  
5.1.0+euler2.8+aach64

在云平台上，环境已经搭建完毕，只需要输入以下指令激活环境

```
source /home/ma-
user/anaconda3/bin/activate
MindSpore
```



## 2 基础成果

选题背景

将实验文件部署在本地以及上传到云平台，文件目录如下

阶段性成果

创新拓展

总结与展望

```
1 ┌── Workspace
2 |   └── lost+found          //自动生成文件夹
3 |   └── code                //Fine tune训练代码
4 |       └── src              //训练脚本调用的配置、数据处理等脚本
5 |       └── preprocessing_dataset.py //预先处理数据集脚本
6 |       └── requirements.txt    //需要的依赖软件包
7 |       └── train.py          //主训练脚本
8 |       └── mobilenetV2.ckpt    //预训练模型
9 |
10 └── kagglecatsanddogs_3367a.zip //数据集
```



选题背景

阶段性成果

创新拓展

总结与展望

## 2 基础成果

### 数据清洗

初始数据集往往存在很多的“脏数据”，降低模型准确率。所以，在实际工程中，往往第一步会进行数据清洗。本实验中，preprocessing\_dataset.py文件主要实现数据集清洗并划分数据集的功能

```
1 (MindSpore) [ma-user work]$cd code
2 (MindSpore) [ma-user code]$python preprocessing_dataset.py ../kagglecatsanddogs_3367a.zip
3 extract dataset
4 extract dataset at /home/ma-user/work/code/dataset/PetImages
5 filter invalid images!
6 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/910.jpg
7 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/445.jpg
8 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/850.jpg
9 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/666.jpg
10 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/23.jpg
11 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/391.jpg
12 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/660.jpg
13 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/936.jpg
14 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Cat/140.jpg
15 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/663.jpg
16 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/50.jpg
17 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/296.jpg
18 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/719.jpg
19 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/565.jpg
20 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/561.jpg
21 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/414.jpg
22 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/522.jpg
23 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/543.jpg
24 Invalid image file, delete /home/ma-user/work/code/dataset/PetImages/Dog/573.jpg
25 filter invalid images done, then split dataset to train and eval
26 final dataset at /home/ma-user/work/code/dataset/PetImages
```

运行preprocessing-dataset文件，删除非图片格式文件，并将数据划分为train和eval用于训练



## 选题背景

## 阶段性成果

## 创新拓展

## 总结与展望

# 2 基础成果

## 训练模型

通过train.py 对预训练mobilenet v2进行微调，再将模型转换格式

class:cat



class:dog



class:dog



训练后分类

class:cat



class:cat



class:cat



```
sh: del: command not found
epoch[14/30], iter[55] cost: 134.843, per step time: 2.452, avg loss: 0.343, acc: 0.980, train acc: 0.998
epoch[15/30], iter[55] cost: 143.149, per step time: 2.603, avg loss: 0.341, acc: 0.985, train acc: 0.999
epoch[16/30], iter[55] cost: 134.089, per step time: 2.438, avg loss: 0.341, acc: 0.975, train acc: 0.999
epoch[17/30], iter[55] cost: 142.364, per step time: 2.588, avg loss: 0.341, acc: 0.985, train acc: 1.000
epoch[18/30], iter[55] cost: 139.609, per step time: 2.538, avg loss: 0.340, acc: 0.975, train acc: 1.000
epoch[19/30], iter[55] cost: 137.125, per step time: 2.493, avg loss: 0.340, acc: 0.990, train acc: 1.000
update best acc: 0.9932659932659934
sh: del: command not found
epoch[20/30], iter[55] cost: 147.932, per step time: 2.690, avg loss: 0.339, acc: 0.975, train acc: 0.999
epoch[21/30], iter[55] cost: 143.839, per step time: 2.615, avg loss: 0.339, acc: 0.975, train acc: 1.000
epoch[22/30], iter[55] cost: 151.530, per step time: 2.755, avg loss: 0.338, acc: 0.985, train acc: 1.000
epoch[23/30], iter[55] cost: 147.527, per step time: 2.682, avg loss: 0.338, acc: 0.985, train acc: 1.000
epoch[24/30], iter[55] cost: 135.690, per step time: 2.467, avg loss: 0.339, acc: 0.980, train acc: 1.000
epoch[25/30], iter[55] cost: 147.175, per step time: 2.676, avg loss: 0.337, acc: 0.985, train acc: 1.000
early stop! the best epoch is 18
train total cost 12.4149 s
export mobilenetv2 MINDIR file at /Users/zhuyu/Desktop/MindSporePetClassification/code/mobilenetv2.mindir
```

训练过程，最终准确率99.3%



## 选题背景

## 阶段性成果

## 创新拓展

## 总结与展望

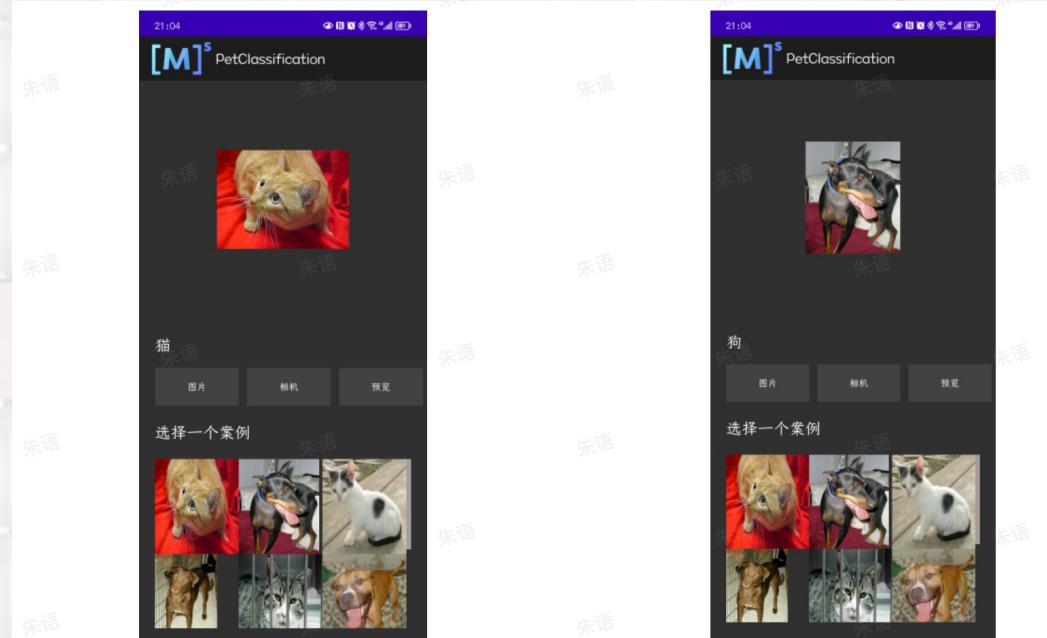
# 2 基础成果

## 移植模型

首先将生成的mobilenetv2.mindir转换成可移植的文件pet.ms

```
1 call .\converter_lite.exe --fmk=MINDIR --modelFile=.\mobilenetv2.mindir  
2 --outputFile=pet
```

在手机上安装app，将pet.ms模型文件导入





选题背景

阶段性成果

创新拓展

总结与展望

3  
创新拓展



选题背景

阶段性成果

创新拓展

总结与展望

# 3 创新拓展

## 一、多任务分类

修改数据集为十分类任务

1. 数据集的获取，我们的数据集来源于Kaggle:Kaggle Animal-10 Dataset

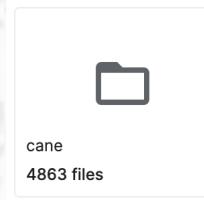
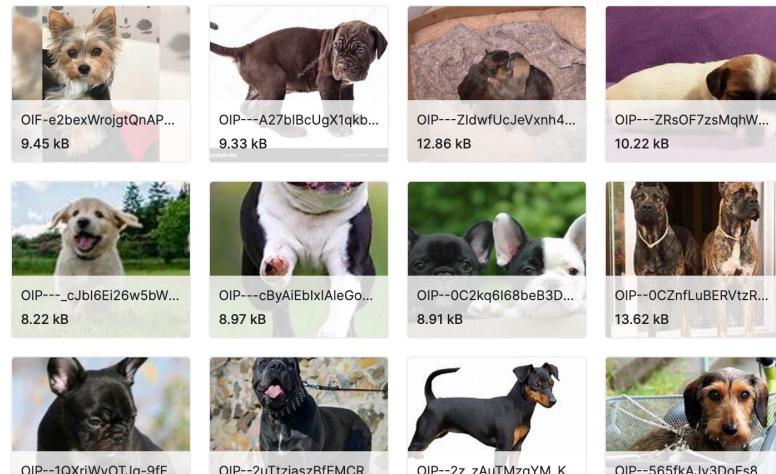
### Animals-10

Data Card Code (130) Discussion (5)

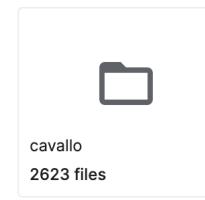
705

New Notebook

cane (4863 files)



cane  
4863 files



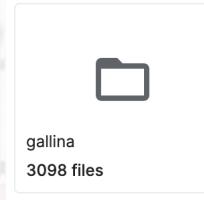
cavallo  
2623 files



elefante  
1446 files



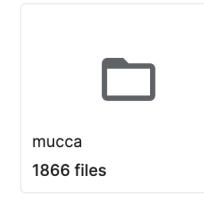
farfalla  
2112 files



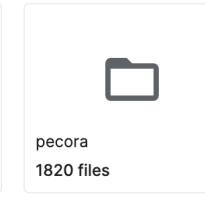
gallina  
3098 files



gatto  
1668 files



mucca  
1866 files



pecora  
1820 files

狗图片

各类数据量



选题背景

阶段性成果

创新拓展

总结与展望

### 3 创新拓展

#### 数据预处理以及模型训练：

调用之前写好的脚本删去错误的文件，再将所有图片转换成规定大小的图片(256)，按照9:1分割训练集和测试集，最后用相同脚本训练。

```
Python ▾
1 import cv2
2
3 import os
4 from os import path
5
6 def image_pre_resize(config, dataset_path):
7     """resize image size to image_height*image_width"""
8     height = config["image_height"]
9     width = config["image_width"]
10
11    image_types = ["train", "eval"]
12    for image_type in image_types:
13        type_path = path.join(dataset_path, f"{image_type}")
14        animal_types = os.listdir(type_path)
15        for animal in animal_types:
16            animal_path = path.join(dataset_path, f"{image_type}/{animal}")
17            image_names = os.listdir(animal_path)
18            for image in image_names:
19                img = cv2.imread(f"{dataset_path}/{image_type}/{animal}/{image}")
20                resized_img = cv2.resize(img, (width, height), interpolation=
cv2.INTER_LINEAR)
21                # write to image
22                cv2.imwrite(f"{dataset_path}/{image_type}/{animal}/{image}",
resized_img)
```

模型准确率会在模型对比部分提到



## 选题背景

## 阶段性成果

## 创新拓展

## 总结与展望

# 3 创新拓展

**APP使用端代码修改：**由于原始APP只能显示“猫”或“狗”，我们这里在Android Studio中对APP代码进行修改，使其能够对十种动物显示进行支持  
由于模型输出未进行归一化，我们在代码中进行Softmax归一化再将结果作为置信度进行输出：

```
1 String[] labels = new String[resultArray.length];
2 double[] scores = new double[resultArray.length];
3 int index = 0;
4 double total = 0;
5 for (String singleRecognitionResult:resultArray) {
6     String[] singleResult = singleRecognitionResult.split(":");
7     double score = Double.parseDouble(singleResult[1]);
8     scores[index] = Math.exp(score);
9     total += scores[index];
10    labels[index] = singleResult[0];
11    index++;
12 }
13 // Softmax 处理
14 for(int i = 0;i < scores.length;i++){
15     scores[i] = scores[i] / total;
16 }
17
18 for (int i = 0; i < resultArray.length;i++){
19     double score = scores[i];
20     String label = labels[i];
21     if (score > 0.5) {
22         recognitionObjectBeanList.add(new RecognitionObjectBean(label,(float) score));
23     }
24 }
25 Collections.sort(recognitionObjectBeanList, new Comparator<RecognitionObjectBean>() {
26     @Override
27     public int compare(RecognitionObjectBean t1, RecognitionObjectBean t2) {
28         return Float.compare(t2.getScore(), t1.getScore());
29     }
30 });

```

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{for } i = 1, \dots, K$$

左图为app代码修改部分，  
右图为softmax公式



选题背景

阶段性成果

创新拓展

总结与展望

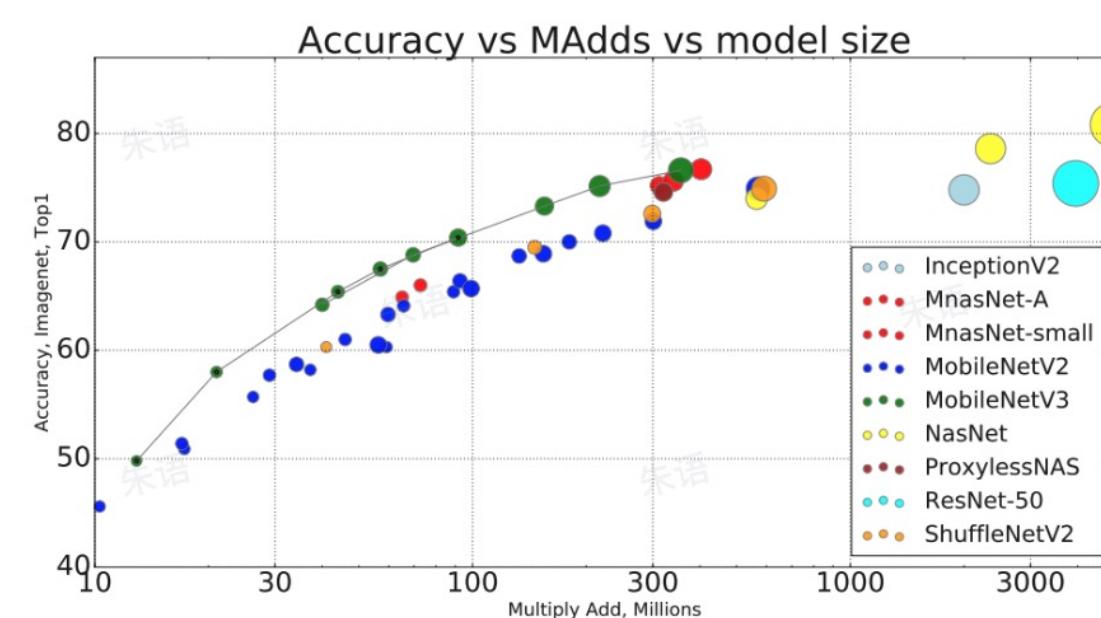
### 3 创新拓展

#### 二、更换分类模型

mobilenet系列网络经过三次迭代。

mobilenetV1创造性的提出了深度可分离卷积，大大减少了模型的计算量； mobilenetV2在第一版的基础上引入线性瓶颈和倒置残差结构。相比前两版， mobilenetV3引入了注意力机制，采用了新的激活函数(h-swish)，重新设计耗时层结构。

我们尝试更换模型为mobilenetV3网络。



在同等计算量下不同模型的规模和准确率的对比（摘自原始论文）



## 选题背景

## 阶段性成果

## 创新拓展

## 总结与展望

# 3 创新拓展

新的模型基于MindCV库搭建。

[MindCV](#)是一个基于[MindSpore](#)开发的，致力于计算机视觉相关技术研发的开源工具箱。它提供大量的计算机视觉领域的经典模型和SoTA模型以及它们的预训练权重和训练策略。

The screenshot shows the official documentation page for MindCV. At the top, there's a navigation bar with links to Home, Installation, Model Zoo, Tutorials, How-To Guides, Reference, Notes, and a search bar. On the right side of the header, it shows the repository name "mindspore-lab/mindcv", the version "v0.2.2", and the number of stars "185" and forks "129".  
  
The main content area has a title "MindCV" and a "Table of contents" sidebar on the right. The sidebar lists several sections: Introduction, Major Features, Model Zoo, Installation, Getting Started, Hands-on Tutorial, Training, Validation, Tutorials, Supported Algorithms, How to Contribute, License, Acknowledgement, and Citation.  
  
The "Introduction" section contains a brief description of MindCV as an open-source toolbox for computer vision research and development based on MindSpore. It highlights features like a series of classic and SoTA vision models, pre-trained weights, training strategies, and auto augmentation methods.  
  
The "Major Features" section includes a bullet point about "Easy-to-Use", stating that MindCV decomposes the vision framework into various configurable components for easy customization.  
  
On the right side of the main content area, there's a sidebar titled "该库支持的模型" (Models Supported by This Library) which lists numerous models with their corresponding GitHub links, such as Big Transfer ResNetV2 (BIT), ConvNeXt, ConViT, DenseNet, DPN, EfficientNet (MBCovNet Family), EfficientNet V2, GhostNet, GoogLeNet, Inception-V3, Inception-ResNet-V2 and Inception-V4, MNASNet, MobileNet-V1, MobileNet-V2, MobileNet-V3 (MBCovNet w/ Efficient Head), NASNet, PNasNet, PVT (Pyramid Vision Transformer), PoolFormer models, RegNet, RepMLP, and RepVGG.



选题背景

阶段性成果

创新拓展

总结与展望

### 3 创新拓展

#### 更新虚拟平台

与前述工作相似，因为MindCV依赖mindspore1.8+版本，我们重新创建ModelArt平台，采用相同硬件参数

(硬件) Ascend: 1\*Ascend910|CPU: 24核 96GB

(软件) mindspore\_1.10.0-cann\_6.0.1-py\_3.7-euler\_2.8.3

#### 模型训练

进入云平台后，我们通过pip安装mindcv库，并将github仓库克隆到平台，修改配置文件后进行训练。

```
Model: mobilenet_v3_small_100
Num batches: 79
Start validating...
batch: 79/79, time: 51.916004s
{'Top_1_Accuracy': 0.887993862677407, 'Top_5_Accuracy': 0.9873417721518988, 'loss': 0.7901543665535843}
```

最终准确率展示



选题背景

阶段性成果

创新拓展

总结与展望

### 3 创新拓展

#### 模型转换

由于生成的是ckpt文件，通过以下代码转换成mindir才能部署到手机端

```
12 def load_model(args):
13     ms.set_context(mode=args.mode)
14     # read num classes
15     num_classes = dataset_eval.num_classes() if args.num_classes is None else args.nui
16     # create model
17     network = create_model(
18         model_name=args.model,
19         num_classes=num_classes,
20         drop_rate=args.drop_rate,
21         drop_path_rate=args.drop_path_rate,
22         pretrained=args.pretrained,
23         checkpoint_path=args.ckpt_path,
24         ema=args.ema,
25     )
26     network.set_train(False)
27
28     return network
29
30
31 if __name__ == '__main__':
32     args = parse_args() # get args
33     model = load_model(args) # load model mobilenetV3 we trained
34     input_tensor = Tensor(np.ones([1, 3, 224, 224]).astype(np.float32)) # define inpu
35     export(model, Tensor(input_tensor), file_name='mobilenetv3', file_format='MINDIR')
```



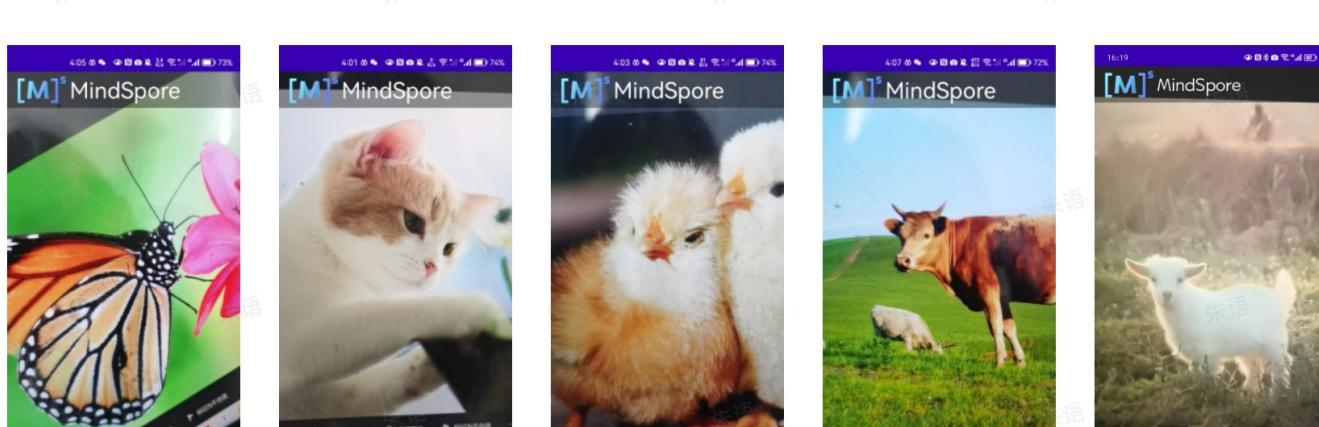
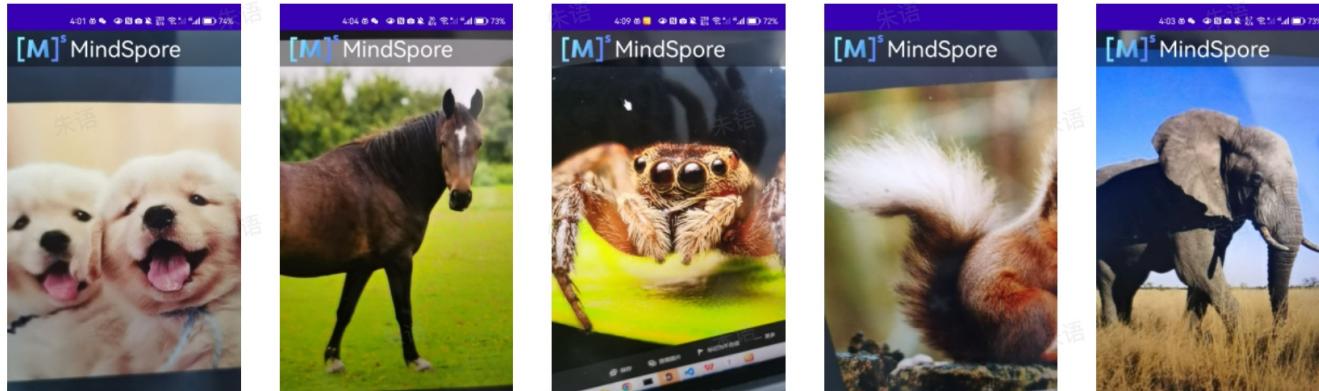
## 选题背景

## 阶段性成果

## 创新拓展

## 总结与展望

# 3 创新拓展—结果展示



十分类结果展示



选题背景

阶段性成果

创新拓展

总结与展望

### 3 创新拓展—结果展示

对比实验：统一模型（mobilenetV2）不同数据集

我们采取了不同的数据集，使用同一模型进行训练，最终试验结果如下表所示：

数据集名称	描述	来源	Acc
动物分类	10 种动物分类	华为平台	99.1
口罩检测	戴/不戴口罩分类	M0 平台	99.9
明星脸识别	明星照片分类	M0 平台	99.0
beans	植物类型识别	Huggingface.co/beans	98.5

基于现有的训练方法和预训练模型



选题背景

阶段性成果

创新拓展

总结与展望

### 3 创新拓展—结果展示

对比实验：不同模型相同数据集

不同针对动物数据集训练相同批（30epoch）并测试，在华为平台上不同模型的分类准确率和响应时间分别是：[←](#)

模型 <a href="#">←</a>	准确率/% <a href="#">←</a>		响应时间/s <a href="#">←</a>	参数量/M <a href="#">←</a>
	top-1 <a href="#">←</a>	top-5 <a href="#">←</a>		
mobilenet v2(1.0) <a href="#">←</a>	85.5 <a href="#">←</a>	99.1 <a href="#">←</a>	50.8 <a href="#">←</a>	3.54 <a href="#">←</a>
mobilenet v3(small_1.0) <a href="#">←</a>	88.8 <a href="#">←</a>	98.7 <a href="#">←</a>	51.9 <a href="#">←</a>	2.55 <a href="#">←</a>

mobilenetv3 在分类 top1 准确率上有显著提升，在 top5 和响应时间方面几乎与 mobilenetv2 保持一致，在模型参数上远远小于 mobilenetv2。[←](#)

基于MindCV库



选题背景

文献综述

理论框架与分析

总结与展望

## 总结与展望

*Summarization & Forseeing*



选题背景

文献综述

理论框架与分析

总结与展望

## 实验总结

## 项目分工

本实验基于MindSpore深度学习框架，通过微调训练网络模型，再将模型转换为可以部署到手机端的ms模型，进而在手机端成功部署。通过此实验，我们对于MindSpore本地环境的搭建、MindSpore框架的应用以及MindSpore Lite框架有了进一步的了解和掌握。

姓名	学号	主要工作	贡献占比
朱语	3200104139	模型拓展、实验测试、报告撰写、展示等	33.33%
沈骏一	3200100259	数据集拓展、实验测试、报告撰写等	33.33%
宋孟炫	3200105210	移动端搭建、实验测试、报告撰写等	33.33%



## 选题背景

## 文献综述

## 理论框架与分析

## 总结与展望

# 实验感想

通过这次实验，我对于人工智能与深度学习在具体任务上的应用有了进一步的认识与操作经验。我们在华为提供的平台上搭建实验模型，并更换数据集、更换模型进行对比实验与分类测试，并成功在本地与移动端两端完成了项目的部署，取得了出色的成效。本次项目经历将在日后更深层次的学习上起到更大作用。

——沈骏一

这次实验让我学习了如何使用华为平台进行模型训练以及移动端的部署。在实验中我尝试使用了新的深度学习工具mindspore和mindcv，多番调试后成功复现，再不同的数据集和网络中运行流畅。这次实验锻炼了我的项目能力、拓展了深度学习网络的知识，也让我对我国人工智能产业发展信心倍增。

——朱语

这次实验我学习了利用华为Mindspore平台进行机器学习的训练，并通过猫狗分类的简单案例带我熟悉了一遍深度学习模型中从数据获取，到模型建立，到训练，部署的全过程及使用的工具，并尝试在一个较为复杂的问题中运用。这次实验让我对于人工智能与机器学习的技术掌握更加深入，运用能力更强。

——宋孟炫



选题背景

文献综述

理论框架与分析

总结与展望

## 参 考 文 献

1. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2019). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510–4520).
2. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1314–1324).
3. Howard, A., & Zhu, M. (2020). MobileNetV3: A search for the best mobile architecture. arXiv preprint arXiv:1905.02244.
4. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946.

THANKS  
恳 请 批 评 指 正

朱语、沈骏一、宋孟炫

2023.07.14