1 滑坡范围

2 户数 具体信息 每个人的信息

3 逃离路线

4 目标：监测数据，位移检测，水位检测，

实时生成图，铜川->推广 全国推广，app，提前预警。

1 精细化

2 逃跑线路 智能化

3 mobile app 岚皋

建立模型，大数据，数据变更，

综合信息的比较，

可实时检测

只看到现象，

我们在测试推理时间时我们从Imagenet的测试集中随机选取图片12000张，其中每张图片测试10次然后取均值得到推理时间。准确率的测试我们数据是Imagenet的整个测试集，共50000张。

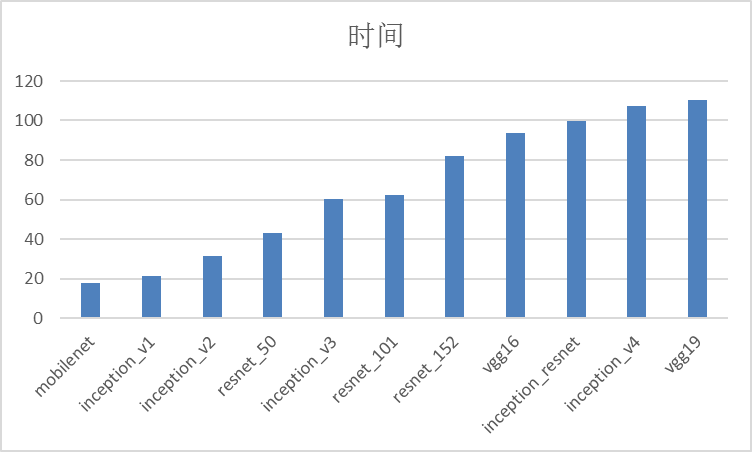
图1 x-axis size/layers label model name

Y-axis accuracy label %

实验整理

1. **图像分类模型的推理时间**

|  |  |
| --- | --- |
| 模型名称 | 推理时间(ms) |
| mobilenet | 18 |
| Inception\_v1 | 21.2 |
| Inception\_v2 | 31.5 |
| Resnet\_50 | 43 |
| Inception\_v3 | 60.3 |
| Resnet\_101 | 62.3 |
| Resnet\_152 | 82.0 |
| Vgg\_16 | 93.9 |
| Inception\_resnet | 99.8 |
| Inception\_v4 | 107.2 |
| Vgg\_19 | 110.3 |



数据集：imagenet

图片数：12000张

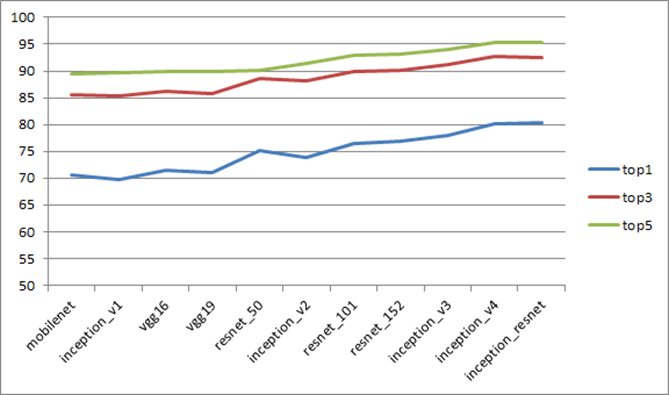
可以看到：对于所有不同类别的图片，在时间上，可以划分为三段20到40,60到80,100到110，其中mobilenet,inception\_v1, Inception\_v2,resnet\_v1\_50在时间上有相当的优势，他们的推理时间集中在20到40之间。

**2．准确率**

我们首先测试的是不同图像分类模型的top1/top3/top5的准确率

数据：imagenet 。数据量：50000张

|  |  |  |  |
| --- | --- | --- | --- |
| 模型名称 | Top1 | Top3 | Top5 |
| mobilenet | 70.7 | 85.5 | 89.6 |
| Inception\_v1 | 69.8 | 85.4 | 89.6 |
| Vgg\_16 | 71.5 | 86.3 | 89.8 |
| Vgg\_19 | 71.1 | 85.8 | 89.8 |
| Resnet\_50 | 75.2 | 88.7 | 90.2 |
| Inception\_v2 | 73.9 | 88.2 | 91.4 |
| Resnet\_101 | 76.4 | 89.8 | 92.9 |
| Resnet\_152 | 76.8 | 90.2 | 93.2 |
| Inception\_v3 | 78 | 91.2 | 94 |
| Inception\_v4 | 80.2 | 92.7 | 95.2 |
| Inception\_resnet | 80.4 | 92.4 | 95.3 |



可以看到的是：不同模型在top1上的差距在百分之十左右，但是到top3后差距在百分之五以内，增长速度变慢，但是top3下的准确率会比top1下平均高百分之10甚至更多。

1在top1下，resnet\_v1\_50较好，尽管时间不是最小的，准确率也不是最高的，但是综合性能较好，如果再继续增加1%的准确率到res\_101，那么时间要增加50%。这样就成本太高。

2在top3,top5下其实他们的准确率幅度变化很小，1%的准确率提高时间需要成倍增加。在top5的情况下，inception\_v3具有更好的性能，时间居中，但准确率几乎最高。

**3.各类模型的操作数在推理时间中所占用的百分比**

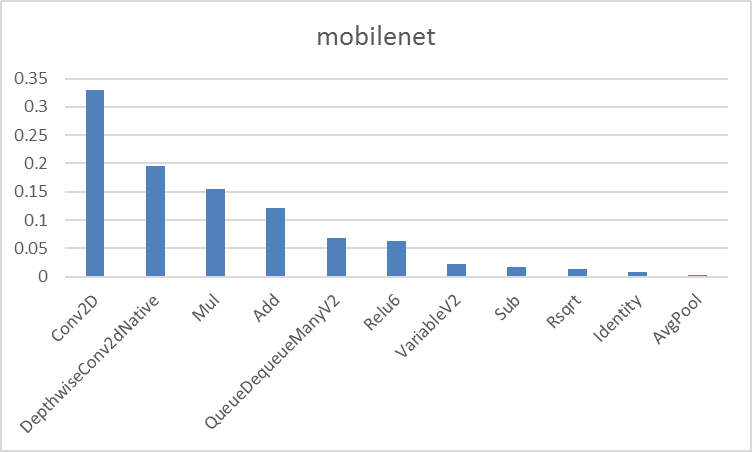
模型所用时间主要是花费在各个类型的操作数上，所以我们对各个模型的操作数在时间上的百分比做了统计。

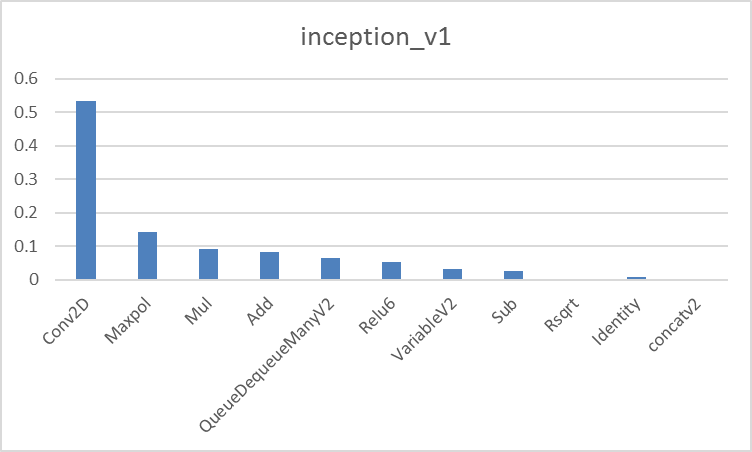
（1）关于图像分类，我们分析可以看到不同类别的操作数比例、类型都比较相似，所以只把有所不同的先画出来。共7个。

数据集：imagenet

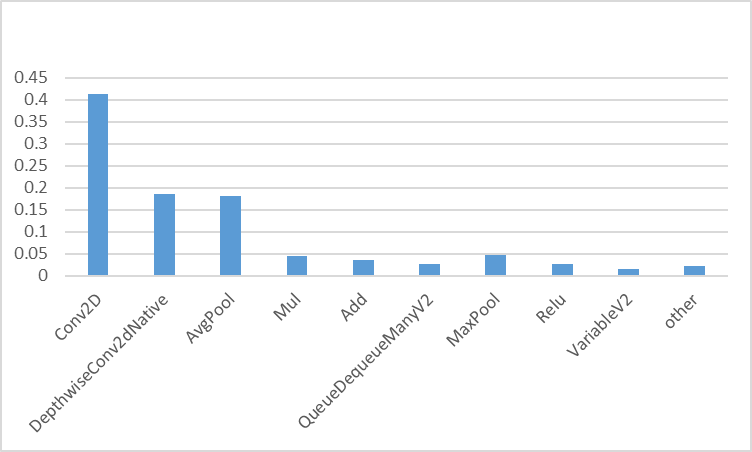
图片数：1000张

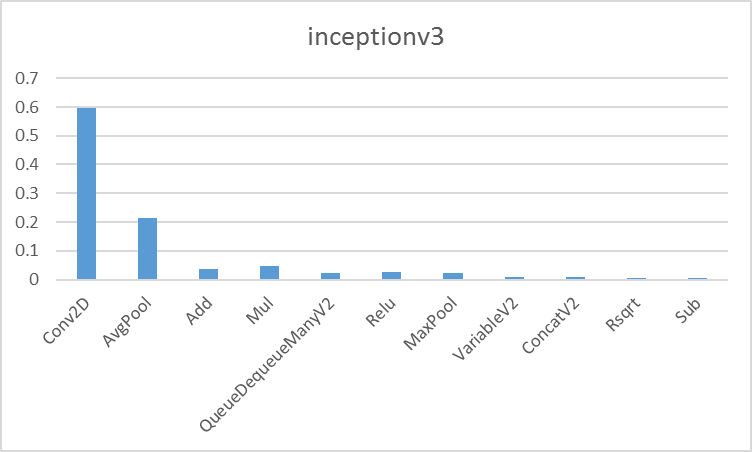
分析：他们在推理过程中的操作数主要可以分为几类：神经网络的组件操作（Conv2D，AvgPool，MaxPool，Relu）,标量运算（Add，Mul,Sub）以及一些队列运算。4种类型的网络主要的操作以卷积操作为主，但根据不同的网络结构，他的百分比有所不同，mobilenet占50%,inceptionnet占60%，resnet可以占到60%到65%，vgg可以达到70%。除卷积操作外，主要还有加法和乘法操作，在fathom里几乎全是全是卷积，我认为应该是训练过程中，时间很长，所以导致他们的百分比很小，而推理过程相对时间缩短很多，所以这些操作所占的百分比有了提升，卷积对于时间的影响相比训练过程有所减弱。不同网络占百分比很小的（2%）一些操作数略有不同。同时，同一种网络他的操作数百分比类似，类型也基本相同，但是推理时间不同，我认为主要差别集中在网络深度上（参数数量）。

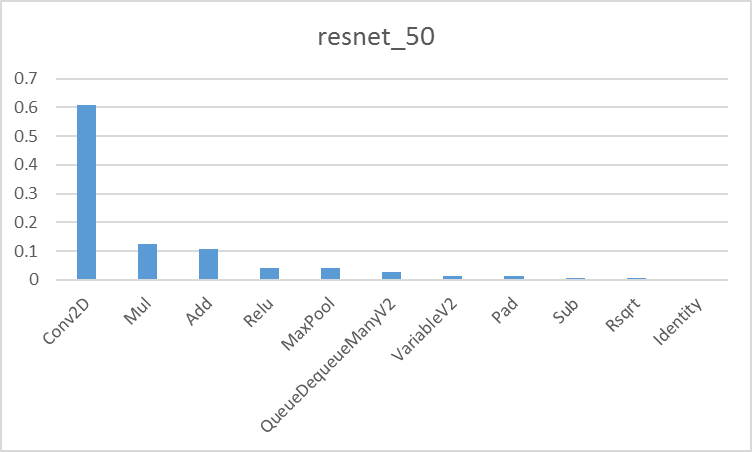


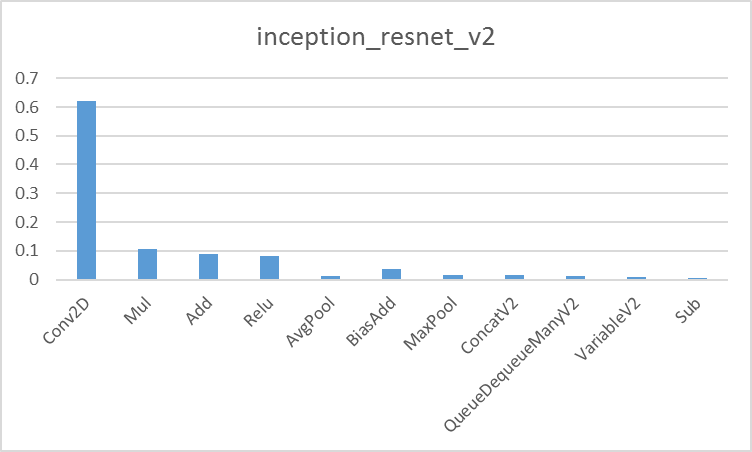


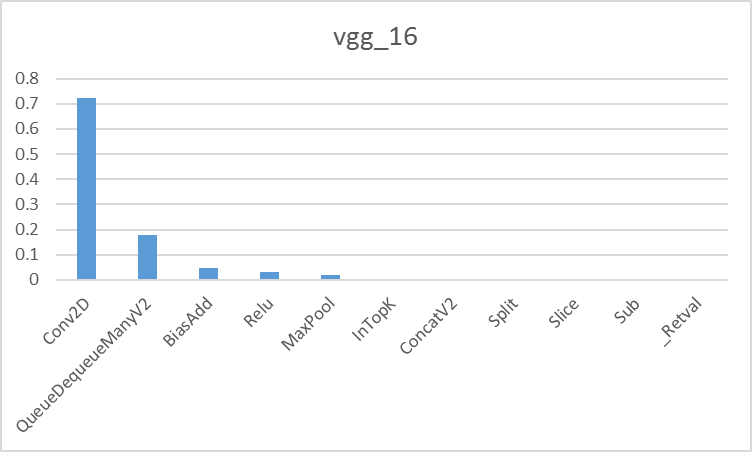
Inceptionv2







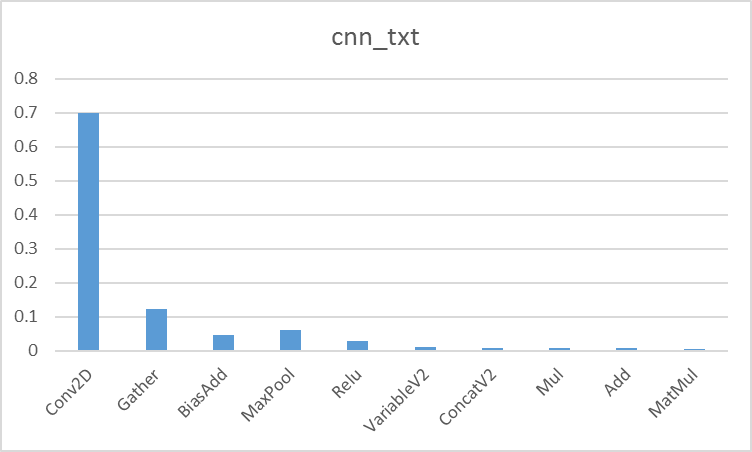




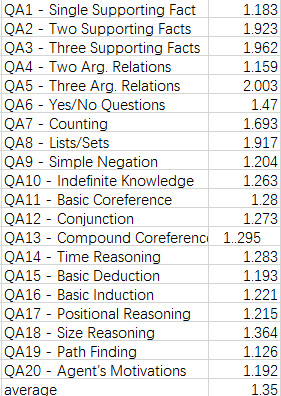
（2）我们又测试了关于一些NLP模型的操作数的统计。选取了不同网络结构的一些典型模型。

1.Convolutional Neural Networks for Sentence Classification。它是Convolutional Neural Networks for Sentence Classification EMNLP 2014这篇文章的一个模型复原，使用卷积网路对文本分类。

数据量有10000条语句，准确率：076。平均推理时间1.18ms(每条句子)。他的操所数主要还是集中在卷积操作，和图片分类类似，但是时间更短，网络结构没有那么复杂。

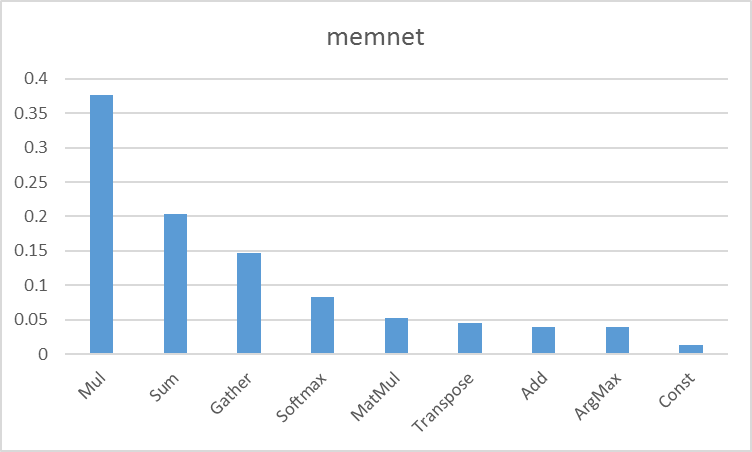


（3）memnet.他是采用End-to-end memory networks. In Advances in Neural Information Processing Systems, NIPS.2015 )的模型复原。采用一种End-to-end模式进行训练，无需知道背景信息中哪些是支撑事实。更适用于实际场景。采用一种神经机器翻译策略的扩展方法。该模型可灵活地解决QA问题。

数据集：bAbI。分别测试了20种不同类型的问答得到他们的推理时间（s）

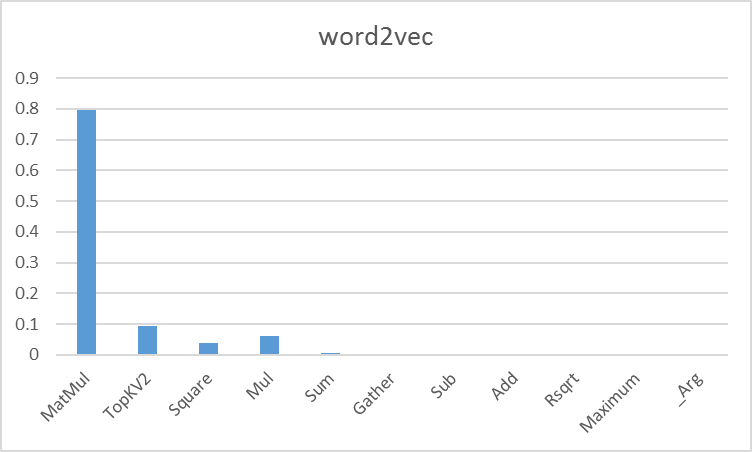
然后进一步测试了他的操所数情况如下图：

他的网络只有三层，是（rnn，dnn）的结构，主要的操作集中在加法和乘法和一些矩阵的运算。

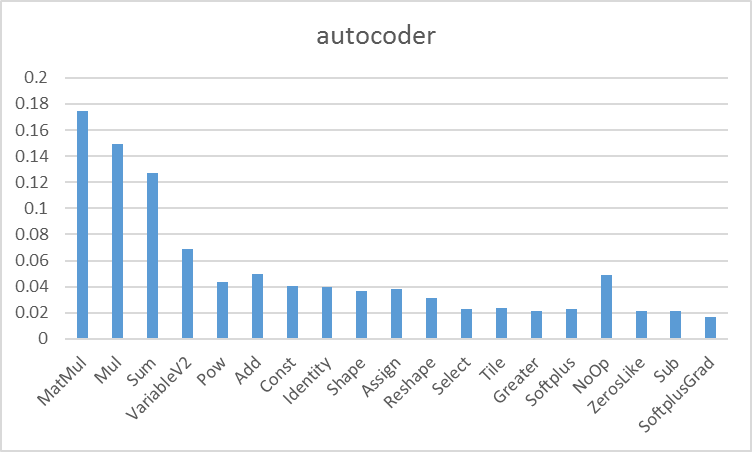


（4）词向量采用了比较经典的embedding算法的例子，是根据Embedding(Efficient Estimation of Word Representations in Vector Space, ICLR 2013.)得到的例子，数据集是根据官方提供的，

具体操作数如下：



（5）自编码器是典型的全连接网络，主要用于模型的特征提取。它是根据Stochastic gradient VB and the variational auto-encoder. In Second International Conference on Learnin Representations, ICLR, 2014.的模型复原。采用数据集是MNIST。主要的操作数集中在乘法和矩阵乘法运算。相对而言他的操作数类型比较多。



通过对这些操作数的测试，我们也看到了不同方向的网络结构他们的操作数类型的相似性是不同的，我们可以根据不同的操作数类型的占用情况分析他们网络结构的差异。

**4．不同类别图片的准确率和推理时间测试**

我们将图片进行分类测试，测试了不同类别的图片在不同模型下的推理时间和不同准确率。

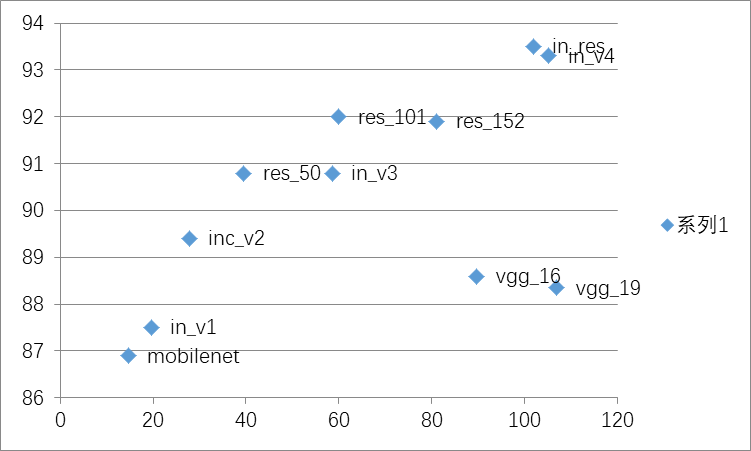
数据包括：我们共测试13550张图片，分为10大类，分别为dog(5900)、bird(2350)、fish(750)、fruit(500)、insect(950)、lizard（550）、monkey、snake(850)、wolf(400)、Monkey(400)、Spider(400)，绘制了7类图片的时间-准确率模型散点图，对比分析模型性能。

这里有两类图片在不同准确率要求下时间和准确率关系，我们可以看到的是：

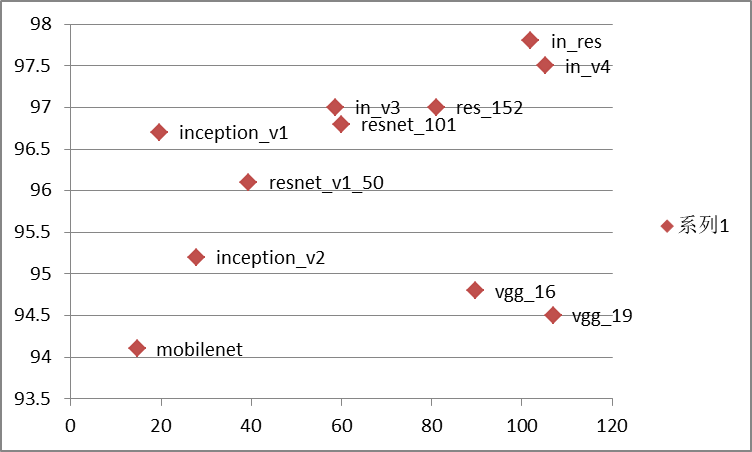
1 不同类别的图像准确率还是有所差别的，比如bird它的准确率远高于平均水平，而snake就很低，bird top1下的准确率甚至高于snake top5下的准确率。然后我们进一步还选取了一些比较数量较小的类别测试，发现准确率主要是受图片类别的影响，和图片的数量关系不大。

2 进一步，如果针对一些类别的图像比如bird,friut这些图像他们的准确率高于平均水平较多，那么我们可以使用简单的模型像mobilenet,inception\_v1等推理时间很短的模型就可以达到比较复杂的网络得到的准确率。就更加方便可以在嵌入式系统上使用。

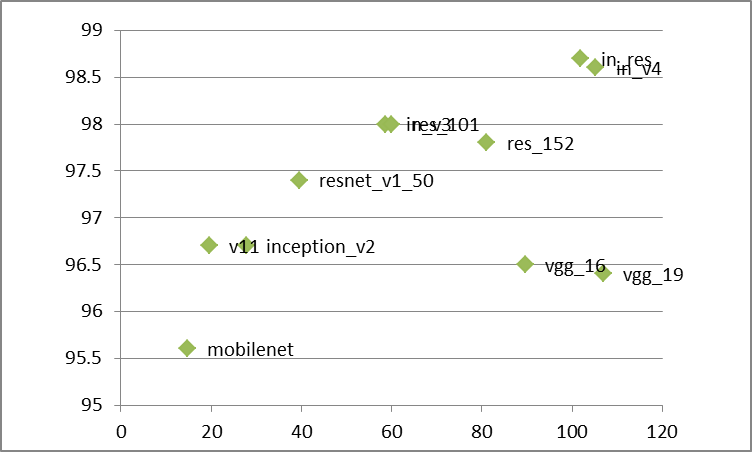
3 一些模型基本是不适合使用在移动端，比如vgg他的推理时间很久，但是准确率却很低，所以我们可以基本不考虑使用。

**Bird：top1准确率及时间（横轴为时间，纵轴为准确率）**

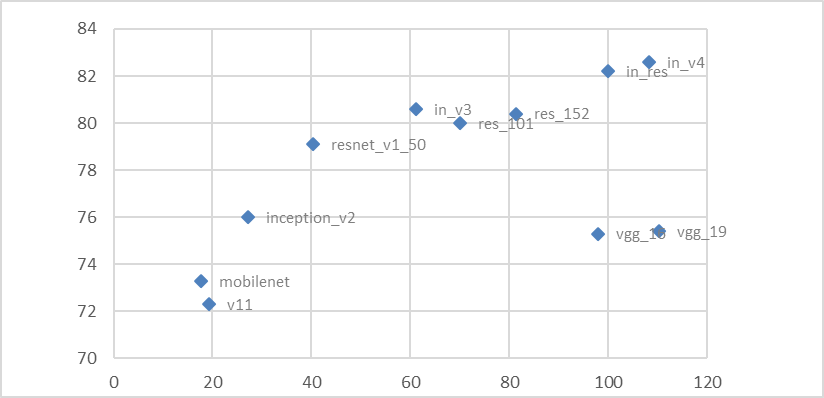
**Bird：top3准确率及时间（横轴为时间，纵轴为准确率）**

****

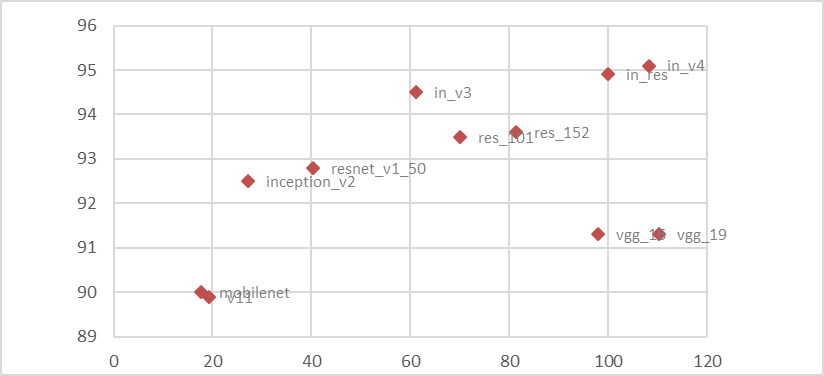
**Bird：top5准确率及时间（横轴为时间，纵轴为准确率）**

****

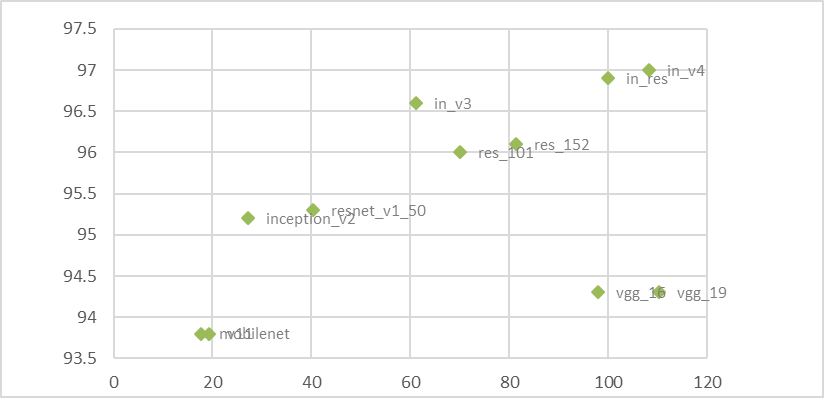
**Dog:top1(5900)**

****

**Top3**

****

**Top5**

****

**5.模型压缩**

由于深度模型普遍较大，我们通过对模型进行量化压缩进一步减少模型大小，使得其可以更方便的部署在移动端**。**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型名称 | 量化前大小 | 量化后大小 | 准确率变换 | 推理时间变化 |
| mobilenet | 17.1M | 4.3M | -23% | -31.25% |
| Inception\_v1 | 26.1M | 6.8M | -3.2% | +25.3% |
| Vgg\_16 | 540.4M | 135.1M | -2.8% | +40% |
| Vgg\_19 | 561.2M | 140.3M | -3.1% | +45% |
| Resnet\_50 | 100.2M | 25.3M | -2.3% | 51% |
| Inception\_v2 | 43.9M | 11.3M | -2% | +41% |
| Resnet\_101 | 174.5M | 44.1M | -2.8% | +158% |
| Resnet\_152 | 235.9M | 58.73M | -3.3% | +180% |
| Inception\_v3 | 92.4M | 24.2M | -2.6% | +280% |
| Inception\_v4 | 167.2M | 42.5M | -1.8% | +198% |

由于mobilenet量化压缩后准确率损失很大，所以我们将mobilenet通过减少宽度，将模型变为更加紧密的小模型从而达到压缩的目的。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 宽度乘数 | 大小 | Top1 | Top2 | Top3 | 推理时间 |
| V1.0 | 17.1M | 0.707 | 0.855 | 0.896 | 18ms |
| V0.75 | 10.5M | 0.618 | 0.838 | 0.880 | 14.7ms |
| V0.5 | 5.4M | 0.641 | 0.805 | 0.854 | 11.3ms |
| V0.25 | 2M | 0.505 | 0.689 | 0.532 | 8.85ms |

Vgg作为一种冗余比较大的模型，我们进一步尝试了模型的通道剪裁来进一步压缩

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 大小变换 | 准确率变化 | 推理时间 |
| 原始模型 | -76% | -3% | -50% |
| 量化+剪裁 |

resnet

|  |  |  |  |
| --- | --- | --- | --- |
| 模型 | 大小变换 | 准确率变化 | 推理时间 |
| 原始模型 | -18.5% | -2% | -20% |
| 剪裁 |

Inception:

