手语检测

Amey Chavan, Shubham Deshmukh, Favin Fernandes

浦那Vishwakarma理工学院电子与电信系

amey.chavan18@vit.edu, shubham.deshmukh18@vit.edu, favin.fernandes18@vit.edu

摘要：随着计算机视觉技术的进步，根据图像的特征对其进行分类已经成为一项巨大的任务和必要性。在这个项目中，我们提出了两种模型，一种是使用ORB和SVM的特征提取和分类，另一种是使用CNN架构。该项目的最终结果是理解特征提取和图像分类背后的概念。经过训练的CNN模型也将用于将其转换为Android开发的tflite格式。

I.介绍

美国手语（ASL）是一种完整的自然语言，与口语具有相同的语言特性，语法与英语不同。ASL是通过手和脸的动作来表达的。它是许多耳聋或重听的北美人的主要语言，也被许多听力正常的人使用。

世界上没有通用的手语。不同的国家或地区使用不同的手语。例如，英国手语（British Sign Language， BSL）与 ASL 是不同的语言，懂ASL的美国人可能听不懂BSL。一些国家在他们的手语中采用了ASL的特征。

没有人或委员会发明了ASL。ASL的确切起源尚不清楚，但有人认为它起源于200多年前当地手语和法国手语（LSF，或法语手语）的混合。今天的ASL包含了LSF的一些元素和原始的当地手语；随着时间的推移，这些元素融合在一起，变成了一种丰富、复杂、成熟的语言。现代ASL和现代LSF是不同的语言。虽然它们仍然包含一些相似的符号，但它们已经不能被对方的使用者理解了。

ASL是一种完全独立于英语的语言。它包含了语言的所有基本特征，有自己的发音、构词法和词序规则。虽然每种语言都有表达不同功能的方式，比如提出问题而不是陈述，但语言在如何做到这一点上是不同的。例如，说英语的人可能会通过提高声音的音调和调整词序来提问；ASL使用者通过扬起眉毛、睁大眼睛和身体前倾来提问。

与其他语言一样，ASL表达思想的具体方式也因ASL使用者自身的不同而不同。除了表达上的个体差异外，ASL还有地域口音和方言；就像某些英语单词在全国不同地区的发音不同一样，ASL在手语的节奏、发音、俚语和使用的标志方面也有地区差异。其他社会因素，包括年龄和性别，也会影响ASL的使用，并导致其多样性，就像口语一样。

手指拼写是ASL的一部分，用于拼写英语单词。在用手指拼写的字母表中，每个字母对应着一个独特的手形。手指拼写通常用于专有名称或表示某物的英语单词。

父母通常是孩子早期习得语言的来源，但对于失聪的孩子来说，额外的人可能是语言习得的模型。一个由已经使用ASL的聋人父母所生的聋儿将开始

学习ASL就像听力正常的孩子从听力正常的父母那里学习口语一样自然。然而，对于没有ASL经验的父母的聋儿来说，语言的习得可能是不同的。事实上，10个天生失聪的孩子中，有9个的父母是听力正常的。一些听力正常的父母会选择向他们的失聪孩子介绍手语。选择让孩子学习手语的听力正常的父母通常会和孩子一起学习手语。父母听力正常的聋人孩子，往往通过聋人同伴学习手语，并变得流利。

为了弥合这些儿童和成人之间的差距，计算机视觉和特征提取技术的使用是很重要的，因为不需要第三方来帮助翻译ASL语言。

因此，建立一个能够识别手语的系统，将会对聋哑人和hard-of-人有所帮助使用现代技术，听力可以更好地沟通。在本文中，我们将通过CNN和ORB的架构来了解它如何对手语进行分类。

II.文献综述

本文介绍[1]实时手语翻译器是促进聋人社区与公众交流的重要里程碑。我们在此提出一个基于皮肤分割和机器学习算法的美国手语（ASL）手指拼写翻译器的开发和实现。我们提出了一种基于颜色信息的人体皮肤自动分割算法。之所以采用YCbCr颜色空间，是因为它通常用于视频编码，并提供了对人类肤色建模的色度信息的有效利用。我们将肤色分布建模为CbCr平面上的二元正态分布。通过对描绘不同种族的人的图像进行模拟，可以说明该算法的性能。然后使用卷积神经网络（CNN）从图像中提取特征，并使用深度学习方法训练分类器来识别手语。

本文介绍了一种基于美国手语的手语识别方法。在这项研究中，用户必须能够使用网络摄像头捕获手势的图像，系统应该预测并显示捕获图像的名称。他们使用HSV颜色算法来检测手势，并将背景设置为黑色。这些图像经过一系列处理步骤，其中包括各种计算机视觉技术，如转换为灰度，扩张和掩模操作。提取的特征是图像的二值像素。他们利用卷积神经网络（CNN）对图像进行训练和分类。他们能够以很高的准确率识别出10个美国手语手势字母。该模型取得了90%以上的显著准确率

在本文[3]中，使用Microsoft Kinect Sensor在杂乱环境中从深度图像中分割手部区域。然后，通过提取和训练图像的特征，将获得的深度图像用于实现监督式机器学习。在这里，通过比较各种方法，可以看出ORB（定向FAST和旋转BRIEF）在精度方面优于其他方法。ORB不受缩放、旋转和光照条件的影响。ORB还与各种分类技术相融合，以获得最佳结果。将该方法应用于ISL 0 ~ 9的图像，并与一些标准数据集进行了比较。使用k-NN分类对ORB进行调优，在ISL数据集上的平均识别准确率为93.26%。

III.Methodolgy

1)数据集

对于这个项目，数据是从Kaggle收集的，它有36个类，包括小写字母A-Z和0-9个数字（每个50张图片）。CNN架构的图像被调整为224x224， ORB的图像被调整为512x512。

图一:美国人的5、4、3标志

2)方法

~~A) 基于ORB和决策树的特征提取与分类:~~

ORB：

该算法是由Ethan Rublee， Vincent Rabaud，Kurt Konolige和Gary R. Bradski在2011年的论文《ORB: SIFT或SURF的有效替代方案》中提出的。正如题目所说，它是SIFT和SURF在计算成本，匹配性能，主要是专利方面的一个很好的替代品。

ORB基本上是FAST关键点检测器和BRIEF描述符的融合，并进行了许多修改以提高性能。首先使用FAST方法找到关键点，然后应用哈里斯角测度方法找到关键点中的前N个点。它还使用金字塔来生成多尺度特征。但有一个问题是，FAST不计算方向。那么旋转不变性呢？作者们提出了以下修改。

它计算具有中心位置角的patch的强度加权质心。矢量从这个角点到质心的方向给出了方向。为了提高旋转不变性，用x和y计算矩，它们应该在半径为r的圆形区域内，其中r是补丁的大小。

对于描述符，ORB使用BRIEF描述符。但是我们已经看到BRIEF在旋转时表现不佳。所以ORB所做的是根据关键点的方向来“引导”BRIEF。对于在位置（xi,yi）上的n个二进制测试的任何特征集，定义一个2×n矩阵，S，它包含了这些像素的坐标。然后利用patch的方向θ，找到它的旋转矩阵，并旋转S得到转向（旋转）版本Sθ。

ORB将角度离散为2π/30（12度）的增量，并构建预先计算的BRIEF模式的查找表。只要关键点方向θ在视图之间是一致的，正确的点集Sθ将被用来计算它的描述符。

BRIEF有一个重要的特性，即每个比特特征有很大的方差和接近0.5的平均值。但是一旦它沿着关键点方向定向，它就失去了这个属性，变得更加分散。高方差使特征更具判别性，因为它对输入的响应是不同的。另一个理想的特性是测试不相关，因为这样每个测试都会对结果有所贡献。为了解决所有这些问题，ORB在所有可能的二进制中运行贪婪搜索

测试来找到那些既有高方差又有接近0.5的均值，以及不相关的。这个结果被称为rbrief。

在描述符匹配方面，采用了改进传统LSH的多探测LSH。ORB比SURF和SIFT快得多，ORB描述符比SURF好。ORB是用于全景拼接等低功耗器件的良好选择。

降维的Kmeans：

K-Means聚类是一种无监督学习算法，它将未标记的数据集分组到不同的聚类中。这里K定义了过程中需要创建的预定义聚类的数量，比如K=2，就会有两个聚类，K=3，就会有三个聚类，以此类推。

它允许我们将数据聚类到不同的组中，并且在不需要任何训练的情况下，可以方便地在未标记的数据集中自己发现组的类别。

它是一种基于质心的算法，其中每个聚类都与一个质心相关联。该算法的主要目的是最小化数据点与其对应的聚类之间的距离之和。

该算法将未标记的数据集作为输入，将数据集划分为k个聚类，并重复该过程，直到没有找到最佳聚类。

该算法通过将数据点聚类到用户给出的没有一个聚类中来降低数据的维数。

决策树分类器：

决策树是一种监督学习技术，既可以用于分类问题，也可以用于回归问题，但大多数情况下更倾向于解决分类问题。它是一种树状结构的分类器，其中内部节点代表数据集的特征，分支代表决策规则，每个叶节点代表结果。

在决策树中，有两个节点，分别是决策节点（Decision Node）和叶节点（Leaf Node）。决策节点用于做出任何决策并具有多个分支，而叶节点是这些决策的输出，并且不包含任何进一步的分支。

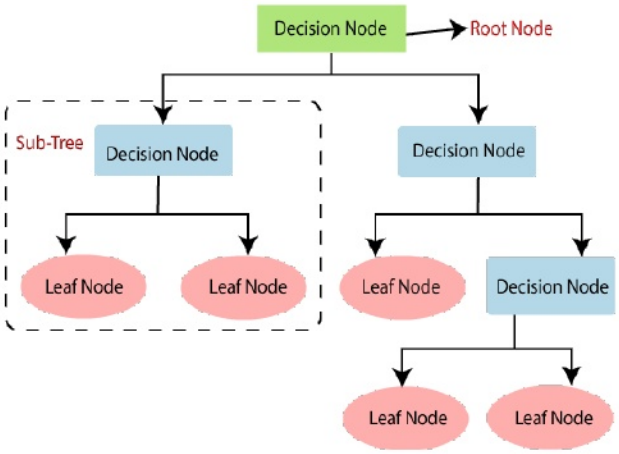
决策或测试是基于给定数据集的特征来执行的。

它被称为决策树，因为与树类似，它从根节点开始，在进一步的分支上扩展，并构建一个树状结构。

为了构建树，我们使用CART算法，即分类回归树算法（Classification and Regression tree）。

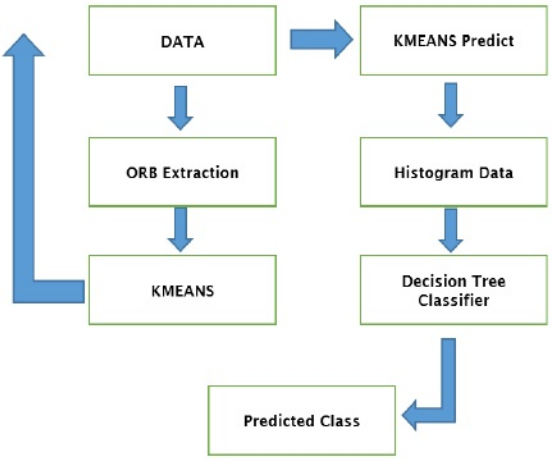
决策树只是提出一个问题，根据答案（是/否），它进一步将树分成子树。

下图解释了决策树的一般结构：



提出的方法：

算法



~~工作:—~~

为了均匀起见，图像首先被调整大小并转换为灰度，特征提取只在灰度图像上进行。提取的特征是ORB特征，整个类的这些特征存储在特征向量中，然后保存在csv文件中。该过程对所有类都运行。

Feature Extraction完成后，对所有类的特征向量进行追加，使用kmeans，拟合5个聚类并保存模型。

然后使用保存的kmeans来预测类的每个图像的kmeans值，这创建了4个特征向量Columns=5。

在为每个类生成kmeans预测特征后，将标签分配给每一行，即所有类的0,1,2,3等。

将这些特征向量附加并馈送到决策树分类器中，通过运行分类器来预测DT分类器的最大深度，直到模型的精度达到最高。

~~B) CNN架构:———~~

卷积神经网络或CNN是一种特殊类型的神经网络，专门用于处理具有网格状拓扑结构的数据，例如图像。神经网络以在各种数据集中发现模式而闻名，卷积尤其适用于图像，因为图像是巨大的数据块，卷积运算可以从中提取有意义的信息，而不会丢失太多信息，并且可以快速执行任务。任何CNN模型都由三个基本层组成：卷积层、池化层、全连接层。在卷积层中，它在两个矩阵之间执行点积，其中一个矩阵是可学习参数的集合，也称为核，另一个矩阵是接受场的受限部分。核用于特征提取，以找出图像中的各种特征，如垂直，水平边缘等，这些核集与输入图像进行卷积。核在空间上比图像小，但更深入。这意味着，如果图像由三个（RGB）通道组成，内核的高度和宽度将在空间上较小，但深度向上扩展到所有三个通道。池化层用于从卷积输出中提取最有用的信息，其他各种命令用于防止过拟合。全连接层用于执行正常的神经网络操作，从前一层的输出中检测模式。与卷积层的各种激活函数（如relu， sigmoid和softmax）一起，用于制定卷积层的输出。所提出的用于手语检测的CNN架构使用4个Conv2d，4个MaxPool2d， 4个Dropout， 1个GlobalAverage Pooling2d和3个Flatten Dense层， 最终使用softmax得到36个多类分类输出。使用Adam优化器作为优化技术，并使用分类交叉熵作为精度度量，模型被拟合并实现了96%的训练精度和92%的测试精度。现在，通过提出的体系结构训练的权重和偏差被保存为keras模型格式(。H5模型)，稍后可以再次转换为tflite模型，并通过移动android应用程序实时检测手语。

C)Android应用

然后将训练好的CNN模型保存在。h5模型中，并转换为tflite格式。的。Tensorflow支持的tflite格式可用于使用Android Studio对图像进行分类。

IV.结果

1) ORB：

特征向量:

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28
0	10	181	61	251	253	253	177	107	159	29	...	122	215	207	215	26	123	210
1	10	181	189	113	185	253	49	59	191	121	...	251	215	207	215	91	103	154
2	205	36	189	247	180	110	121	154	123	255	...	235	98	207	255	119	68	192
3	65	235	10	10	120	157	102	231	174	1	...	198	53	116	21	18	105	80
4	32	237	83	8	91	201	167	231	14	5	...	64	61	53	36	16	43	87
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5785	0	157	150	185	151	237	176	103	190	3	...	18	247	141	54	30	75	152
5786	194	160	132	216	203	225	56	10	158	19	...	194	211	29	18	218	239	144
5787	0	175	150	185	147	224	176	231	62	19	...	2	255	45	182	30	67	209
5788	130	134	129	153	211	224	184	71	62	23	...	130	215	13	54	154	235	144
5789	2	156	50	184	87	165	185	102	190	5	...	18	247	221	54	26	106	148

173755 rows × 32 columns

减少尺寸:

	0	1	2	3	4	5
0	74	52	89	50	81	1
1	149	86	91	61	92	1
2	35	44	105	57	81	1
3	34	69	97	49	84	1
4	49	30	74	67	64	1
...	...	...	...	...	...	...
25	3	13	10	19	14	27
26	23	15	28	37	14	27
27	49	15	38	28	35	27
28	51	44	35	35	69	27
29	36	25	33	48	59	27

780 rows × 6 columns

决策树分类器准确率：20%

2) CNN

模型简介:

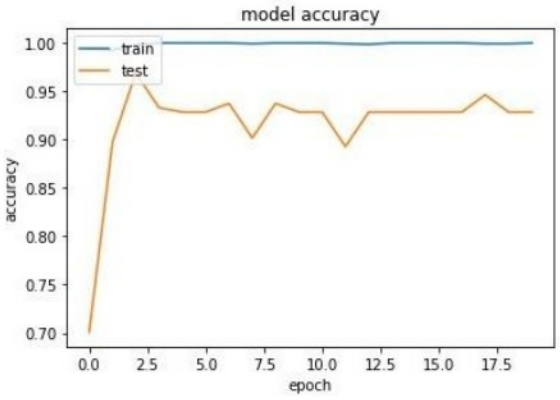
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_1 (Conv2D)	(None, 112, 112, 128)	73856
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_2 (Conv2D)	(None, 56, 56, 256)	295168
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 256)	0
batch_normalization (BatchNo	(None, 28, 28, 256)	1024
flatten (Flatten)	(None, 200704)	0
dropout (Dropout)	(None, 200704)	0
dense (Dense)	(None, 1024)	205521920
dense_1 (Dense)	(None, 36)	36900

Total params: 205,930,660
Trainable params: 205,930,148
Non-trainable params: 512

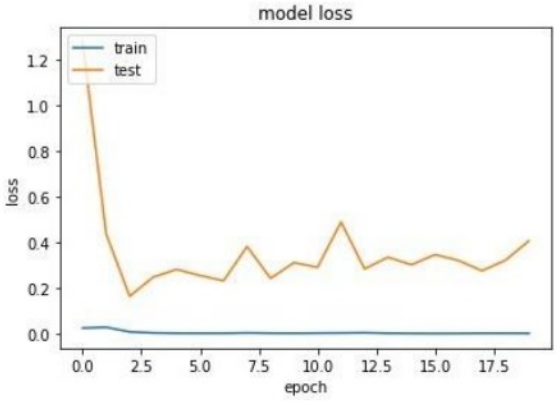
每个Epoch的训练精度：

```
Epoch 6/20
48/48 [==============================] - 29s 593ms/step - loss: 0.0016 - accuracy: 1.0000 -
Epoch 7/20
48/48 [==============================] - 28s 579ms/step - loss: 0.0014 - accuracy: 1.0000 -
Epoch 8/20
48/48 [==============================] - 29s 592ms/step - loss: 0.0040 - accuracy: 0.9986 -
Epoch 9/20
48/48 [==============================] - 28s 587ms/step - loss: 0.0020 - accuracy: 1.0000 -
Epoch 10/20
48/48 [==============================] - 28s 578ms/step - loss: 0.0011 - accuracy: 1.0000 -
Epoch 11/20
48/48 [==============================] - 28s 576ms/step - loss: 0.0018 - accuracy: 1.0000 -
Epoch 12/20
48/48 [==============================] - 28s 579ms/step - loss: 0.0024 - accuracy: 0.9999 -
Epoch 13/20
48/48 [==============================] - 28s 575ms/step - loss: 0.0048 - accuracy: 0.9982 -
Epoch 14/20
48/48 [==============================] - 28s 575ms/step - loss: 0.0023 - accuracy: 1.0000 -
Epoch 15/20
48/48 [==============================] - 28s 576ms/step - loss: 0.0010 - accuracy: 1.0000 -
Epoch 16/20
48/48 [==============================] - 28s 573ms/step - loss: 3.9363e-04 - accuracy: 1.00
Epoch 17/20
48/48 [==============================] - 27s 566ms/step - loss: 0.0015 - accuracy: 1.0000 -
Epoch 18/20
48/48 [==============================] - 27s 568ms/step - loss: 0.0014 - accuracy: 0.9993 -
Epoch 19/20
48/48 [==============================] - 27s 563ms/step - loss: 0.0014 - accuracy: 0.9993 -
Epoch 20/20
48/48 [==============================] - 27s 564ms/step - loss: 8.9692e-04 - accuracy: 1.00
```

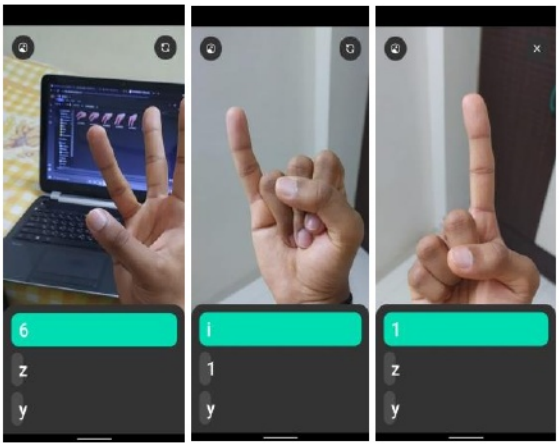
模型精度:



模型损失:



3)Android Application



V.结论

ORB特征提取的计算速度更快，但分类模型的准确率很低。另一方面，CNN在ORB方面提供了很好的结果。CNN模型还可以转换为许多其他格式，以便在Web， Android和Flutter程序中实现用户训练的模型。
至少，当涉及到特征提取和分类时，CNN技术具有优势。

Refernces:

[1] S. Shahriar et al., "Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning," TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 1168-1171, doi: 10.1109/TENCON.2018.8650524

[2] Mehreen Hurroo , Mohammad Elham, 2020, Sign Language Recognition System using Convolutional Neural Network and Computer Vision, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 12 (December 2020)

[3] Gangrade, Jayesh & Bharti, Jyoti & Mulye, Anchit. (2020). Recognition of Indian Sign Language Using ORB with Bag of Visual Words by Kinect Sensor. IETE Journal of Research. 1-15. 10.1080/03772063.2020.1739569

[4] Ss, Shivashankara & S, Dr.Srinath. (2018). American Sign Language Recognition System: An Optimal Approach. International Journal of Image, Graphics and Signal Processing. 10. 10.5815/ijigsp.2018.08.03

[5] Mehreen Hurroo , Mohammad Elham, 2020, Sign Language Recognition System using Convolutional Neural Network and Computer Vision, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 12 (December 2020)

[6] R. A. Pranatadesta and I. S. Suwardi, "Indonesian Sign Language (BISINDO) Translation System with ORB for Bilingual Language," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), 2019, pp. 502-505, doi: 10.1109/ICAIIT.2019.8834677.

[7] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or

SURF," *2011 International Conference on Computer Vision*, 2011, pp. 2564 2571, doi: 10.1109/ICCV.2011.6126544.

[8] F. Yang, "An Extended Idea about Decision Trees," *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2019, pp. 349-354, doi: 10.1109/ CSCI49370.2019.00068.

[9] S. Lu *et al*., "Clustering Method of Raw Meal Composition Based on PCA and Kmeans," *2018 37th Chinese Control Conference (CCC)*, 2018, pp.90079010, doi:10.23919/ChiCC.2018.848282

[10] A. S. Nikam and A. G. Ambekar, "Sign language recognition using image based hand gesture recognition techniques, " 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-5, doi: 10.1109/ GET.2016.7916786.

[11] Sahoo, Ashok & Mishra, Gouri & Ravulakollu, Kiran. (2014). Sign language recognition: State of the art. ARPN Journal of Engineering and Applied Sciences. 9. 116-134.