数字音视频分析实验报告（一）

姓名 金家耀

学号 1193210320

班级 人工智能 2103

学院 人工智能与计算机学院

指导老师 王锐

2023 年 9 月 21 日

目录

数字音视频分析实验报告（一）

1 实验要求

1. 给定的 dataset 中包含了 4 个手势类别，每个类别由 100 张 RGB 图片组成；
2. 各位同学首先需要将所有的 RGB 图片转换成 HSV 类型，然后从中提取亮度（灰度）信息；接着，对上述灰度图像施加直方图均衡化操作，以期降低图片中的光照因素对后续分类结果的影响；
3. 对于上述多分类问题（4 类），请同学们自行选择一种机器学习算法，如 Linear Discriminant Analysis (LDA)、Kernel Discriminant Analysis (KDA)、Sparse Representation (SR)、Dictionary Learning (DL) 等，并结合简单的分类器（K 近邻、支持向量机等）对 2）中处理后的数据进行训练和测试。此外，也可选用简单的神经网络模型。
4. 编程语言不限：MATLAB、Python、C 等均可以。

2 数据集部分展示

图 1: 四种手势数据集

如图 1所示，数据集中包含四种类型的手势图片：左偏、剪刀手、分指、和右偏手势。每张手势图片的尺寸均为 320x240，并且拍摄质量非常清晰。值得注意的是，观察每种手势图片时，可以注意到每张图像的手势角度略有变化，但总体上噪声较少。（然而，为了提高机器学习和神经网络模型的性能，可能需要引入一些具有噪声的图像样本。

3 图像预处理

3.1 RGB 转 HSV 并提取亮度（灰度）信息

RGB 颜色转化成 HSV 颜色公式如下：

1. 将 RGB 颜色值标准化到 [0, 1] 范围内，即将每个颜色通道的值除以 255（最大颜色通道值）：

$$R' = \frac{R}{255}$$
$$G' = \frac{G}{255}$$
$$B' = \frac{B}{255}$$

2. 计算最大通道和最小通道的值，以确定色相（H）：

$$C_{\max} = \max(R', G', B')$$
$$C_{\min} = \min(R', G', B')$$
$$\Delta = C_{\max} - C_{\min}$$

3. 计算色相（H）：

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \cdot \left(\dfrac{G' - B'}{\Delta} + 0 \right) & C_{\max} = R' \\ 60^\circ \cdot \left(\dfrac{B' - R'}{\Delta} + 2 \right) & C_{\max} = G' \\ 60^\circ \cdot \left(\dfrac{R' - G'}{\Delta} + 4 \right) & C_{\max} = B' \end{cases}$$

4. 计算饱和度（S）：

$$S = \begin{cases} 0 & C_{\max} = 0 \\ \dfrac{\Delta}{C_{\max}} & \text{otherwise} \end{cases}$$

5. 计算亮度（V）：

$$V = C_{\max}$$

如图 2所示，为数据集第一张图有 RGB 转成 HSV 后分离的三通道展示图。

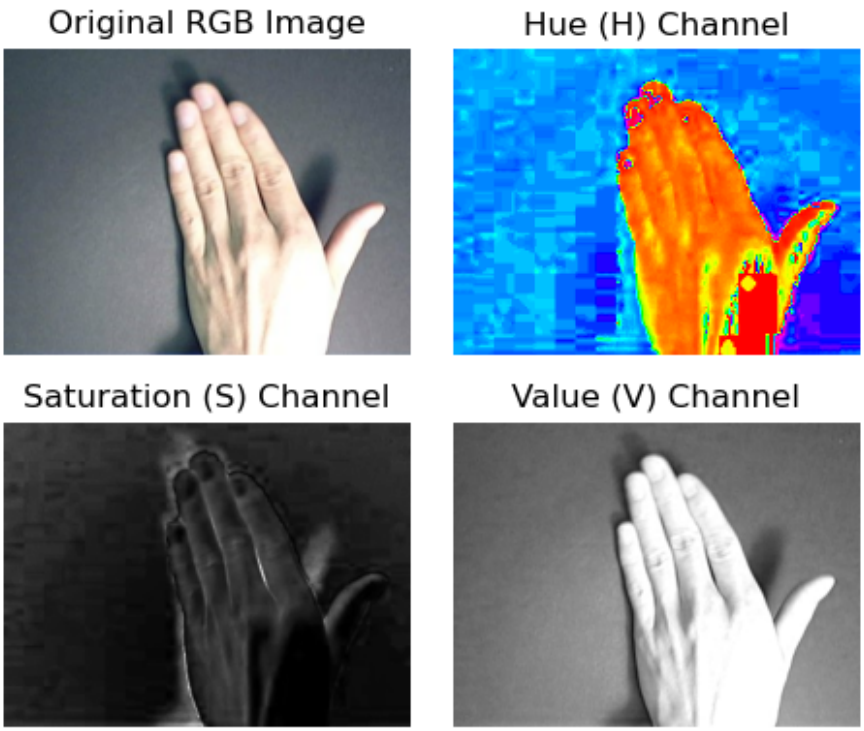


图 2: 原图以及 HSV 三通道展示图

3.2 直方图均衡化

给定一个输入图像 $I(x,y)$，其中 (x,y) 表示图像的像素坐标，以及 L 为像素灰度级别的总数（通常为 256，对应于 8 位图像），直方图均衡化可以通过以下步骤来实现：

1. 计算输入图像的直方图：$H(i) = n_i/N$，其中 $H(i)$ 是像素值 i 的频率分布，n_i 是图像中像素值为 i 的像素数量，N 是图像总像素数量。

2. 计算累积分布函数（CDF）：$C(i) = \sum_{j=0}^{i} H(j)$，其中 $C(i)$ 是像素值 i 的累积分布函数。

3. 将 CDF 线性拉伸到 0 到 L-1 的范围内：$S(i) = \frac{C(i)-C_{\min}}{1-C_{\min}} \cdot (L-1)$，其中 $C_{\min}$ 是 CDF 的最小非零值。

4. 使用新的像素值替换原始图像中的每个像素：$I_{\text{equalized}}(x,y) = S(I(x,y))$，其中 $I_{\text{equalized}}(x,y)$ 是均衡化后的图像，$I(x,y)$ 是原始图像。

通过这个过程，图像的对比度得以显著增强，更广泛的亮度级别被有效利用，从而显著提升了图像的视觉质量。如图 3所示，这是使用数据集中的第一张手势图进行的直方图均衡化后的直方图。从直方图的对比度明显提升的图示中，可以清晰地观察到，均衡化后的图像像素范围更加广泛。这有助于将原始图像中紧密聚集的像素值分散开，为后续的机器学习算法和神经网络提供更好的输入数据。

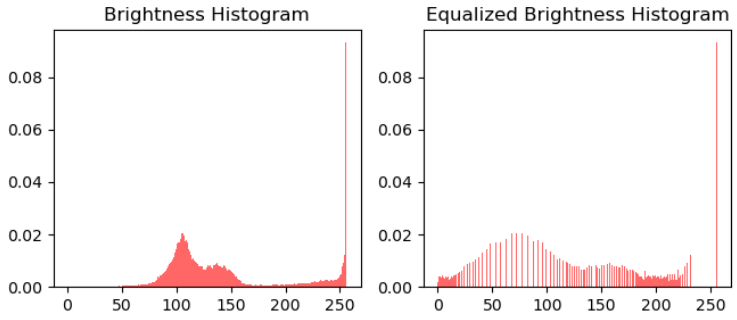


图 3: 直方图均衡化

4 模型选择——机器学习

4.1 介绍

4.1.1 LDA 算法

LDA 的全称是 Linear Discriminant Analysis（线性判别分析），是一种 supervised learning。有些资料上也称为是 Fisher’s Linear Discriminant，因为它被 Ronald Fisher 发明自 1936 年。LDA 是在目前机器学习、数据挖掘领域经典且热门的一个算法，据我所知，百度的商务搜索部里面就用了不少这方面的算法。

1. 二分类问题

假设数据集 $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$，现在假设是二分类问题，则 $y_i \in \{0, 1\}$。假设第一类样本个数为 N_1，第二类样本个数为 N_2。定义 μ^j 为第 j 类样本的均值，Σ^j 为第 j 类样本的方差。计算公式如下：

$$\begin{cases} \mu^j = \dfrac{1}{N_j}\Sigma_{x \in X_j} x \\ \Sigma^j = \displaystyle\sum_{x \in X_j} (x - \mu^j)(x - \mu^j)^T \end{cases}$$

据我个人理解，分类的好坏在于类间散度和类内散度，所以我们最终需要使得类间散度与类内散度的比值越大越好。现假设我们的投影直线为向量 ω，则对于任意一个样本 x_i，它在直线 ω 上的投影为 $\omega^T x_i$，则对于两类均值 μ^j，其在直线上的投影为 $\omega^T \mu^j$，则方差为 $\omega^T \Sigma^j \omega$，最终我们得到目标函数：

$$argmax_{\{\omega\}} J(\omega) = \frac{||\omega^T \mu^0 - \omega^T \mu^1||_2^2}{\omega^T \Sigma^0 \omega + \omega^T \Sigma^1 \omega} = \frac{\omega^T (\mu^0 - \mu^1)(\mu^0 - \mu^1)^T \omega}{\omega^T (\Sigma^0 + \Sigma^1) \omega}$$

令 $S_b = (\mu^0 - \mu^1)(\mu^0 - \mu^1)^T$，$S_w = \Sigma^0 + \Sigma^1$，则公式变为：

$$argmax_{\{\omega\}} J(\omega) = \frac{\omega^T S_b \omega}{\omega^T S_w \omega}$$

根据瑞利商公式，$J(\omega^*)$ 的最大值为 $S_w^{-\frac{1}{2}} S_b S_w^{-\frac{1}{2}}$ 的最大特征值，而 ω^* 为该特征值对应的特征向量。

2. 多分类问题

由于该实验做的是四分类问题，所以需要进一步搞清楚多分类 LDA 算法的计算过程。同样地，假设数据集 $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$，则 $y_i \in \{1, 2, ..., k\}$，其中 k 为样本类别个数。每个类别的样本个数、样本均值、样本方差与二分类 LDA 算法一致。

由于该问题是多分类问题，所以投影的不再是一条直线，而是一个超平面 W，设该超平面的维度为 d，其基向量为 $(\omega_1, \omega_2, ..., \omega_d)$。

由二分类问题推广，我们的目标函数变为：

$$argmax_{\{W\}} J(W) = \frac{W^T S_b W}{W^T S_w W}$$

其中，$S_b = \sum_{j=1}^{d} (\mu^j - \mu)(\mu^j - \mu)^T$，$\mu$ 为所有样本的均值，$S_w = \sum_{j=1}^{k} \Sigma^j$。

由于目标函数是一个矩阵优化问题，故无法从二分类问题中推广，则这里会有个小小的目标函数的松弛（个人觉得很精辟）：

$$argmax_{\{W\}} J(W) = \frac{\Pi_{diag} W^T S_b W}{\Pi_{diag} W^T S_w W} = \frac{\Pi_{i=1}^{d} \omega_i^T S_w \omega_i}{\Pi_{i=1}^{d} \omega_i^T S_b \omega_i} = \Pi_{i=1}^{d} \frac{\omega_i^T S_w \omega_i}{\omega_i^T S_b \omega_i}$$

可以观察，化简最后的式子其实也是广义瑞利熵，最终的 W^* 其实就是 $S_w^{-\frac{1}{2}} S_b S_w^{-\frac{1}{2}}$ 的最大 d 个特征根所对应的特征向量。

4.1.2 SVM

支持向量机（Support Vector Machine，SVM）是一种用于分类和回归问题的机器学习算法。它的主要目标是找到一个最佳的超平面，将数据集中的样本分为不同的类别。SVM 主要概念如下：

- **超平面**：在二维空间中，超平面是一条直线；在三维空间中，它是一个平面。在高维空间中，它是一个超平面。超平面的方程通常表示为：$\omega \cdot x + b = 0$，其中，w 是法向量（perpendicular vector）或权重向量（weight vector），b 是偏差（bias），x 是数据点。

- **间隔**：间隔是指离超平面最近的数据点到超平面的距离。SVM 的目标是最大化这个间隔。

- **支持向量**：支持向量是离超平面最近的那些数据点，它们对于定义超平面起关键作用，因为它们决定了间隔的大小和超平面的位置。

- **硬间隔与软间隔**：在实际问题中，数据往往不是线性可分的，这时可以使用软间隔 SVM。硬间隔 SVM 要求所有数据点都正确分类，而软间隔 SVM 允许一些数据点被错误分类，但同时会引入一个惩罚项，以平衡间隔的大小和错误分类的数量。

对于二分类问题，模型的目标是最大化：$\frac{2}{|w|}$，约束条件：$y_i(w \cdot x_i + b) \geq 1$ 对于所有$i = 1, 2, \ldots, n$，其中，y_i 是数据点 x_i 的类别标签，w 是法向量，b 是偏差，n 是数据点的数量。这个问题可以通过拉格朗日乘数法来求解，得到最优的权重向量 w 和偏差 b。对于多分类问题，可以使用一对多（One-vs-Rest）策略，将每个类别分别与其他类别进行比较，构建多个二分类 SVM。

4.2 训练与测试

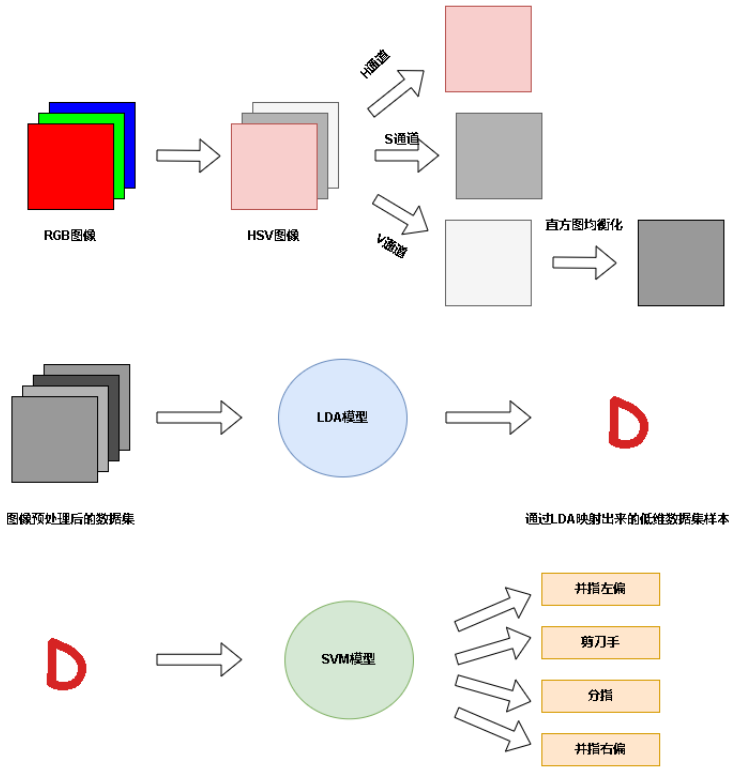
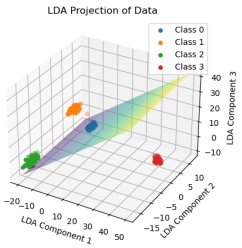


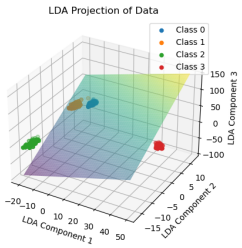
图 4: 机器学习算法流程图（自画）

如图 4所示，为机器学习算法流程图，包含图像预处理过程、LDA 降维过程和 SVM 模型分类过程。图像预处理过程在第三节中有所提到，这里就不多做赘述。使用预处理过后的图像将其扁平化并使用 LDA 模型进行降维，由实验结果显示，降维之后的维度为 3 维，由后续的可视化中可以看出降维的效果较好。将降维过后的样本特征输入 SVM 中，进行一对多的决策平面拟合，由于该实验中数据集的类别是 4 类，故决策平面应为 6 个。

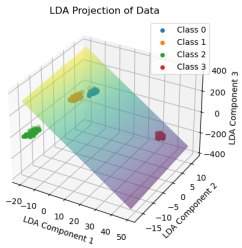
如图 5所示，画出了数据集在 LDA 映射下的样本空间以及通过 SVM 分类得到的 6 个决策平面。观察图，可以得出一对多的决策平面表现良好。为了表明图像预处理的有效性，也做了直接往模型中输入原始图片的灰度图来进行预测分类，得到以下表格：



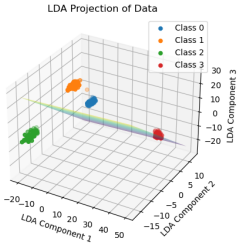
(a) 类 1 与类 2 的决策平面



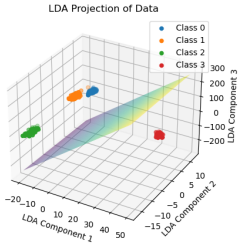
(b) 类 1 与类 3 的决策平面



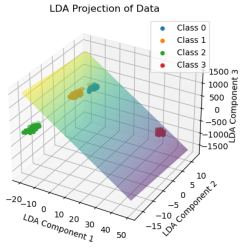
(c) 类 1 与类 4 的决策平面



(d) 类 2 与类 3 的决策平面



(e) 类 2 与类 4 的决策平面



(f) 类 3 与类 4 的决策平面

图 5: 数据集在 LDA 映射空间以及六个决策平面展示

表 1: 手势数据集在 LDA+SVM 组合模型上的表现

数据集类别	Precision	Recall	F1-Score
经过预处理	1.0	1.0	1.0
未经过预处理	1.0	1.0	1.0

5 模型选择——神经网络

5.1 介绍

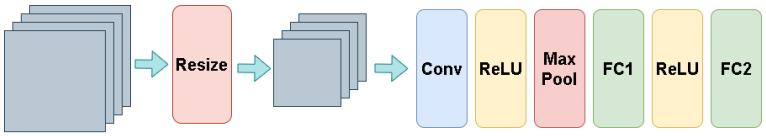


图 6: 神经网路结构可视化

由于上一节中机器学习算法效果太好了，没有一项指标能让我看出图像预处理的效果。而使用神经网络训练会有一个迭代的过程，我们可以观察迭代的过程来观察图像预处理是否有效。

而对于图像分类数据集，我想到的是卷积提取特征再使用全连接进行分类，最后交叉熵损失函数进行反向传递，用来拟合标签。

5.1.1 网络结构说明

如图 5.1 所示，为神经网路结构可视化图。

- 卷积层 1：第一层是一个卷积层，输入通道数为 1，输出通道数为 16，使用 3x3 的卷积核进行卷积操作。
- ReLU 激活函数：在卷积层后面，使用了 ReLU 激活函数，引入非线性性质。
- 最大池化层：紧接着卷积层和激活函数层，有一个最大池化层，用于减小特征图的尺寸。
- 全连接层 1：池化层的输出被展平成一维向量，然后输入到全连接层 1 中，输出节点数为 64。
- ReLU 激活函数：全连接层 1 后，再次经过 ReLU 激活函数。
- 全连接层 2：最后一个全连接层有 64 个输入节点和 4 个输出节点，是输出层，用于分类问题。

Code Listing 1: 神经网络模型的 Python 代码

```python
class Model(nn.Module):
    def __init__(self):
        super(Model, self).__init__()
        self.conv1 = nn.Conv2d(1, 16, kernel_size=3, padding=1)
        self.relu1 = nn.ReLU()
        self.pool1 = nn.MaxPool2d(kernel_size=2, stride=2)
        self.fc1 = nn.Linear(16 * 40 * 30, 64)  # 减少全连接层的节点数
        self.relu2 = nn.ReLU()
        self.fc2 = nn.Linear(64, 4)  # 输出层，假设有4个类别

    def forward(self, x):
        x = self.pool1(self.relu1(self.conv1(x)))
        x = x.view(x.size(0), -1)
        x = self.fc1(x)
        x = self.relu2(x)
        x = self.fc2(x)
        return x
```

5.1.2 交叉熵损失函数（Cross-Entropy Loss Function）

交叉熵损失函数，通常简称为交叉熵（Cross-Entropy），是在分类问题中广泛使用的损失函数之一。它用于度量模型的输出与实际标签之间的差异，是一种衡量分类模型性能的指标。

对于二分类问题，交叉熵损失的公式如下：

$$H(y,p) = -\left(y\log(p) + (1-y)\log(1-p)\right)$$

其中：

- $H(y,p)$ 是交叉熵损失。
- y 是实际的类别标签，取值为 0 或 1，表示样本属于正类别（1）或负类别（0）。
- p 是模型的预测概率，表示样本属于正类别的概率。

对于多分类问题，交叉熵损失的公式如下：

$$H(y,p) = -\sum_{i=1}^{N} y_i \log(p_i)$$

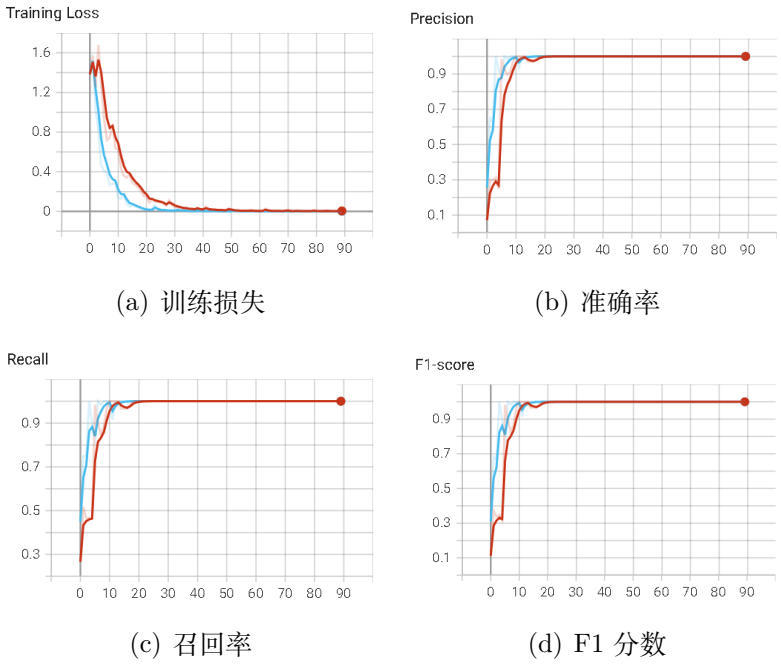
其中：

- $H(y,p)$ 是交叉熵损失
- y 是实际的类别标签的独热编码，表示样本属于哪个类别。
- p 是模型的预测概率向量，表示每个类别的预测概率。
- N 是类别的总数。

交叉熵损失的目标是使模型的预测概率分布尽可能接近实际的类别分布。当模型的预测与实际情况一致时，交叉熵损失为 0，表示模型的性能最佳。当模型的预测与实际情况不一致时，交叉熵损失增加，表示模型的性能差。

5.2 训练

当进行神经网络训练之前，我首先将原始数据集分成了训练集和测试集，采用了 8:2 的比例划分。这个步骤的目的在于确保我们有一个独立的数据集，可以用来评估训练好的模型性能，以便进行有效的模型评估。在数据预处理方面，我对图像进行了一系列操作，以改善模型的训练效果。为了验证数据预处理的效果，如图 7所示，我记录了四个关键指标随着训练步数的变化，分别是训练损失、准确率、召回率和 F1 分数。在训练指标的对比图中，蓝色曲线代表了经过数据预处理后的训练效果，而红色曲线代表



(a) 训练损失　　(b) 准确率

(c) 召回率　　(d) F1 分数

图 7:　原始输入与预处理输入对模型训练的影响

了未经过预处理的效果。观察这些图表，我们可以明显看出数据集经过预处理后的显著优势。

在训练损失方面，蓝色曲线收敛得更快，这表明模型更快地学习到了数据的特征和模式。而在准确率、召回率和 F1 分数方面，尽管两者最终都达到了 1.0，但经过数据预处理的数据集在训练过程中明显优于未经过预处理的数据集。这说明数据预处理有助于模型更快地达到高性能水平，并且更容易稳定在高性能状态。

综上所述，数据集的预处理在神经网络训练中起到了关键作用，不仅加速了模型的收敛速度，还提高了模型的性能表现。这对于实现高效的深度学习模型至关重要。

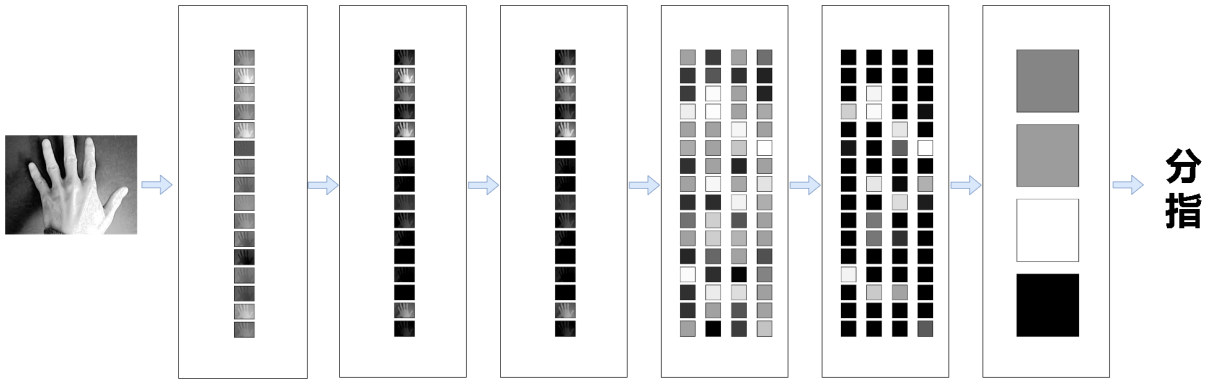


图 8: 模型中间结果可视化

5.3 测试

如图 8，我使用了第三类“分指”的一个样本来进行模型中间结果可视化，从原始样本分别进行卷积、池化、激活层、两个全连接层之后进行分类预测。从图中看出，最后四个色块代表模型对于该样本归一化之后的概率，颜色由黑到白是从 0.0 到 1.0，图中第三个色块为白色，故模型预测该样本属于第三类——分指。

在训练过程中和训练结束后我分别进行了测试集图像的预测，如图 9所示，为一次预测结果。由于模型性能，测试时显示的预测结果均与原样本一致，模型表现较好。

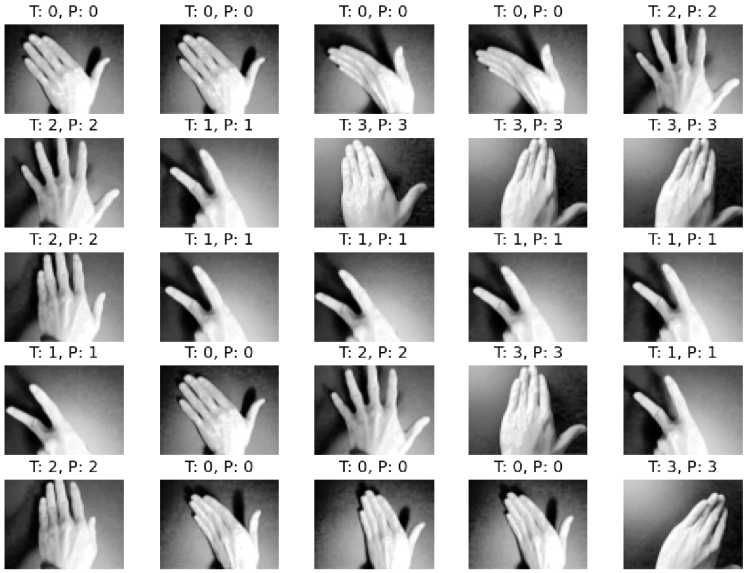
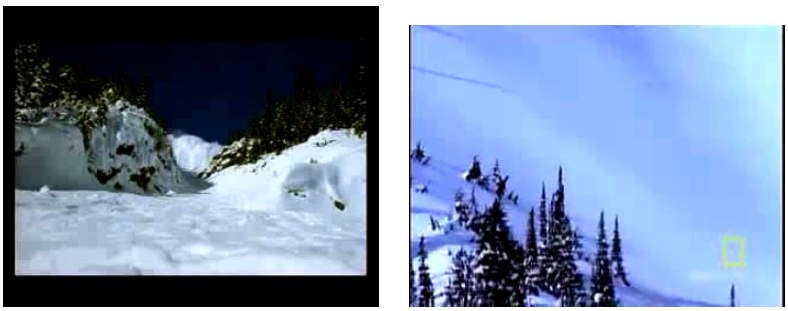


图 9: 模型训练预测结果

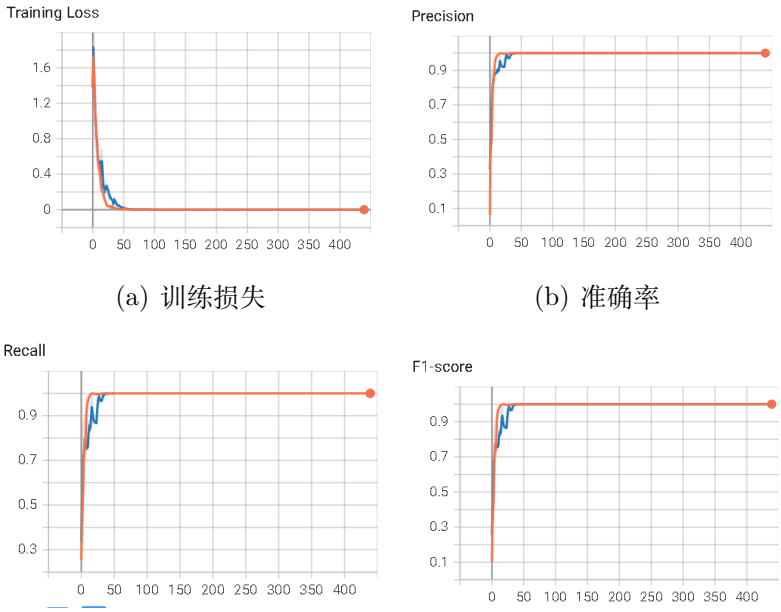
6 数据集拓展

前面说到，手势数据集较小且噪声较少，很难看出图像预处理前后的差距。于是使用老师提供的有 2000 张数据样本的数据集作为拓展数据集。



(a) 样本 1　　(b) 样本 2

图 10: MDSD_subset 数据集展示



(a) 训练损失　　(b) 准确率

(c) 召回率　　(d) F1 分数

图 11: 原始输入与预处理输入对模型训练的影响

6.1 介绍

MDSD_subset 数据集共有 4 个类别，每个类别包含 2 个子类（其中一个子类中有 285 张图片，另一个子类中有 215 张，共计 500 张）。如图 10所示，为雪崩样本的两个子集展示图片。

6.2 数据预处理

由于四大类的各两个子类都分别位于两个文件夹中，我首先将两个子类随机合并为一个类别。观察到数据集中的图片尺寸不同，我利用 cv.resize 将每一张图片都统一为 $(60, 80)$。随后，和手势数据集的操作一致，将 RGB 图像转化为 HSV 并提取亮度信息。

6.3 LDA+SVM

运用相同的 LDA 与 SVM 组合模型，将输入改为新的数据集，得到以下结果：

表 2: MDSD_subset 数据集在 LDA+SVM 组合模型上的表现

数据集类别	Precision	Recall	F1-Score
经过预处理	1.0	1.0	1.0
未经过预处理	1.0	1.0	1.0

6.4 神经网络

我同样利用 tensorboard 工具记录了神经网路在该数据集中的表现，结果如图 11所示，从结果同样可以看出图像预处理之后的数据集可以是使更快地收敛。

6.5 总结

由上述结果可得，虽然图像预处理前后训练的模型都表现较好，但图像预处理能够使模型更快地收敛，更快地找到梯度下降的策略。同时，我观察到虽然同一个类中两个子类表达的是同一个意思，但其图片表征仍然存在一定的差异，包括颜色、纹理等，但无论是机器学习还是神经网路，都能够从这两个表征不同的子类中提取出相同的特征使得模型得到收敛。

7 思考、总结与拓展

7.1 为什么实验中不使用灰度图而是使用 HSV 的 V 通量？

从两次神经网络的对比中可以看出，使用 HSV 图像的 V 通道要比直接使用彩色图的灰度图来进行训练效果更好，我不禁提出疑问，两者代表的不都是亮度信息吗？从网络中得到计算灰度图的计算公式如下：

$$Gray = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

RGB 转成的 V 通道（亮度通道）和直接由 RGB 转成的灰度图之间有一些区别：

- 计算方式：
  - V 通道：V 通道是通过将 RGB 图像转换为 HSV 色彩空间，然后提取亮度通道得到的。HSV 转换考虑了颜色信息和亮度信息，因此 V 通道的值受到颜色信息的影响。
  - 灰度图：直接由 RGB 图像转换为灰度图像是一种常见的方法，通常使用加权平均法将三个通道的值合并为一个灰度值。这种方法忽略了颜色信息，仅考虑亮度。
- 信息保留：
  - V 通道：保留了原始 RGB 图像的颜色信息，但更强调亮度。这意味着 V 通道可以包含一些颜色变化的信息。
  - 灰度图：只保留了亮度信息，完全忽略了颜色信息。这意味着灰度图中不包含颜色信息。

- 亮度强调:
  - V 通道：V 通道的值受到颜色信息的影响，因此在存在颜色变化的区域，V 通道的亮度可能会有所不同。
  - 灰度图：灰度图是根据亮度计算得出的，不考虑颜色。因此，即使存在颜色变化，灰度图中的亮度也会保持一致。
- 用途:
  - V 通道：通常用于颜色图像处理中的一些任务，其中需要同时考虑颜色和亮度信息，例如图像分割、对象检测等。
  - 灰度图：通常用于需要仅考虑亮度而不考虑颜色的任务，例如边缘检测、图像压缩等。

总的来说，V 通道和灰度图都有其特定的用途，具体取决于您的任务需求。如果需要保留颜色信息并考虑亮度，V 通道可能更适合；如果只关注亮度，那么灰度图可能更合适。

7.2 一些个人想法

我参与的大创项目涉及图像处理领域，具体是文本生成视频。在立项阶段，我们的导师建议我们关注当时备受关注的"stable diffusion" 技术，这也成为了我们项目的核心方向。最终，我们确定了文本生成视频作为研究方向，并已成功发表了一篇普通的 EI 会议论文。但是在做科创的过程中，我认为我对于图像处理、图像融合、视觉方向是比较有兴趣的，但是我依然对图像处理一些理论知识缺乏完整地理解，只能做到应用，我希望以后我能更加注重与基础理论地学习，能够把理论在应用中不断拓展。