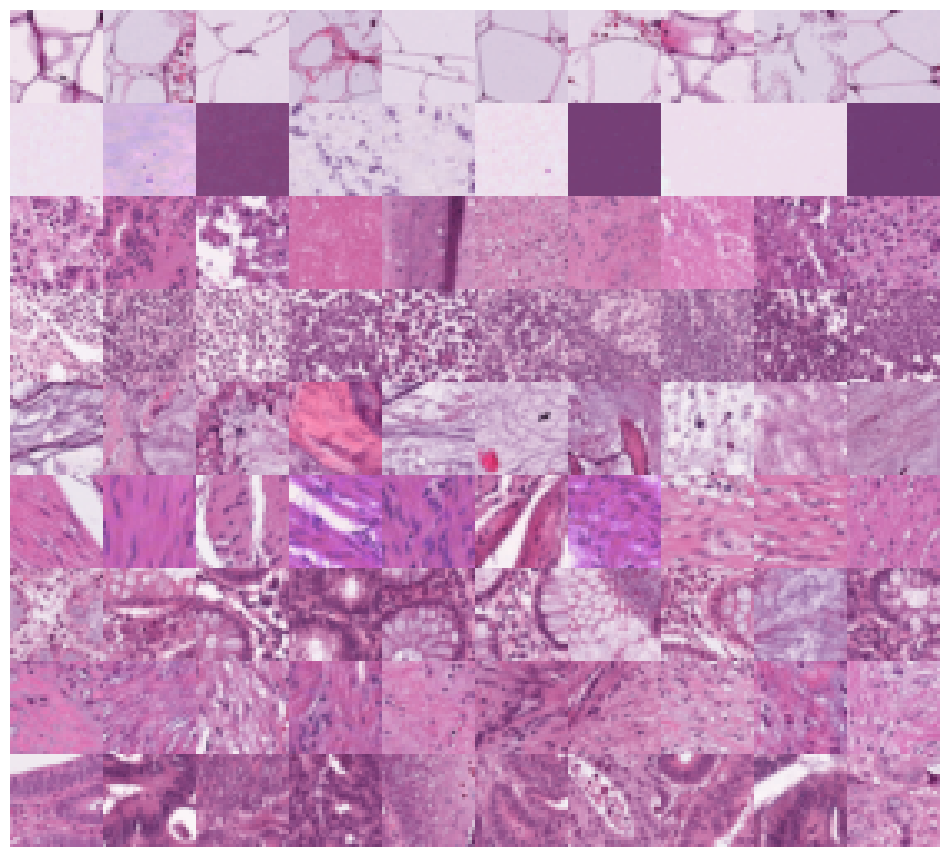
# Introduction:

# Data exploration:

原始样本包含training 89996,validation 10004, test 7180, 由学校提供的版本仅包含training 32000条和test 8000条。

所有的样本被分为9种类别，class0-class8.

**Data Modality:** Colon Pathology，

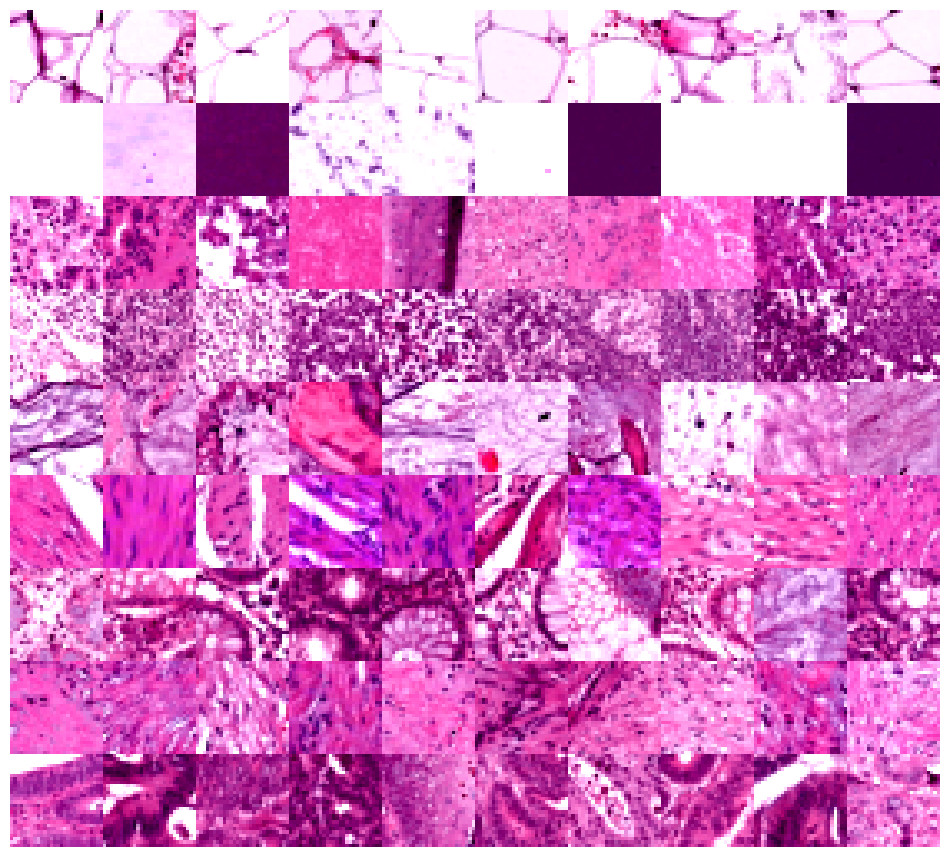


图为试输出的数据图片，从上到下分别为Label0-label8, label0 为adipose, label1为background,label2 为debris, label3 为lymphocytes， label4为mucus, label 5为 smooth muscle， label6为normal colon mucosa，label7 为 cancer-associated stroma，label8为 colorectal adenocarcinoma epithelium。

通过观察输出图片了解到，background类别内存在显著的颜色差异，可能造成捕捉特征的困难; smooth muscle 类别有些样本的颜色特别的鲜艳; normal colon mucosa、cancer-associated stroma和colorectal adenocarcinoma epithelium类别之间，有些样本无法通过人眼的观察分辨出其区别，这些样本的特征并不明显，可能对模型的训练造成阻碍。

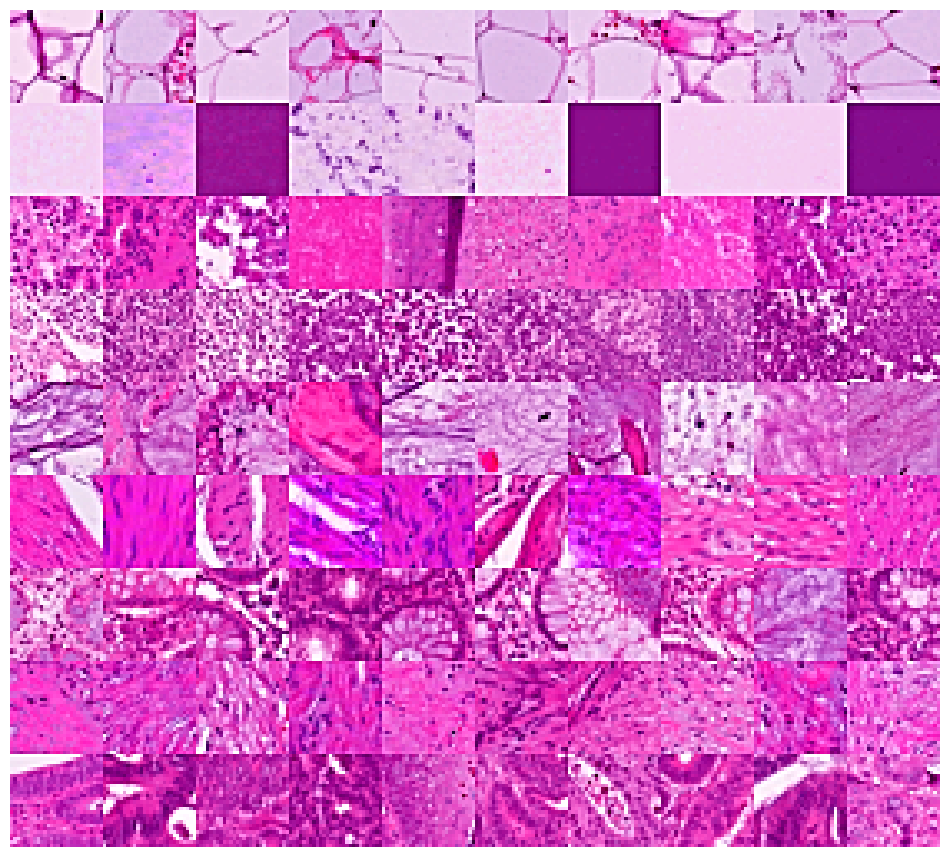
# Preprocessing

1. 首先考虑到样本图片的主要色调差异不大，考虑通过强化对比度来强化其特征。

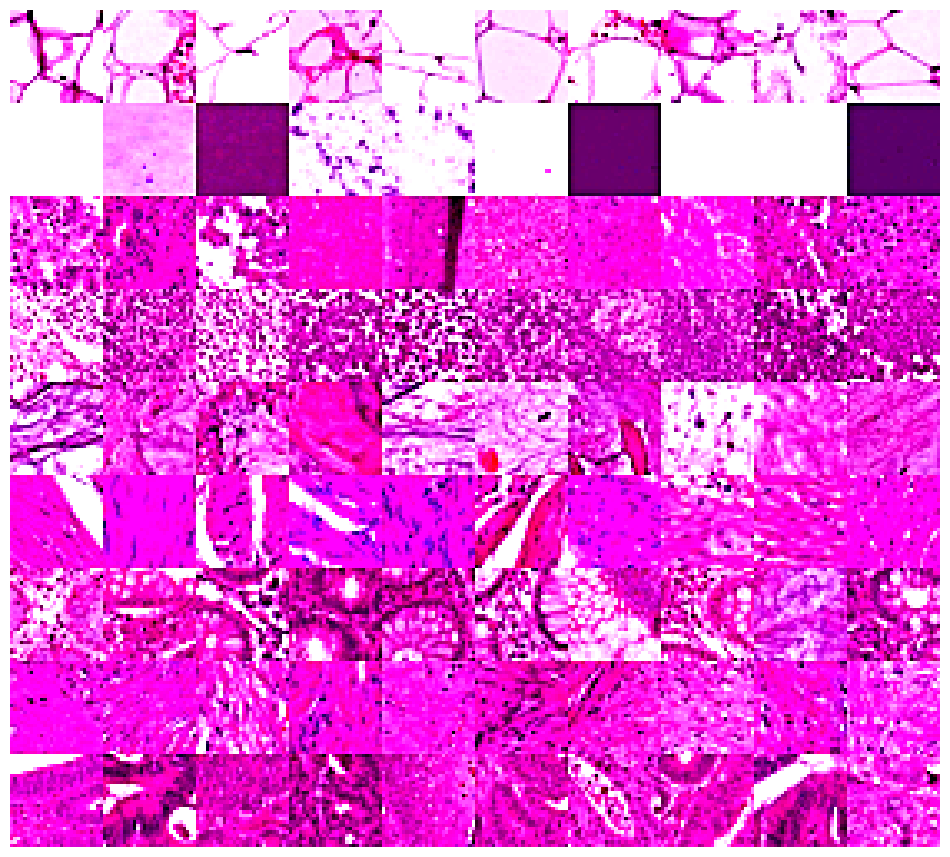


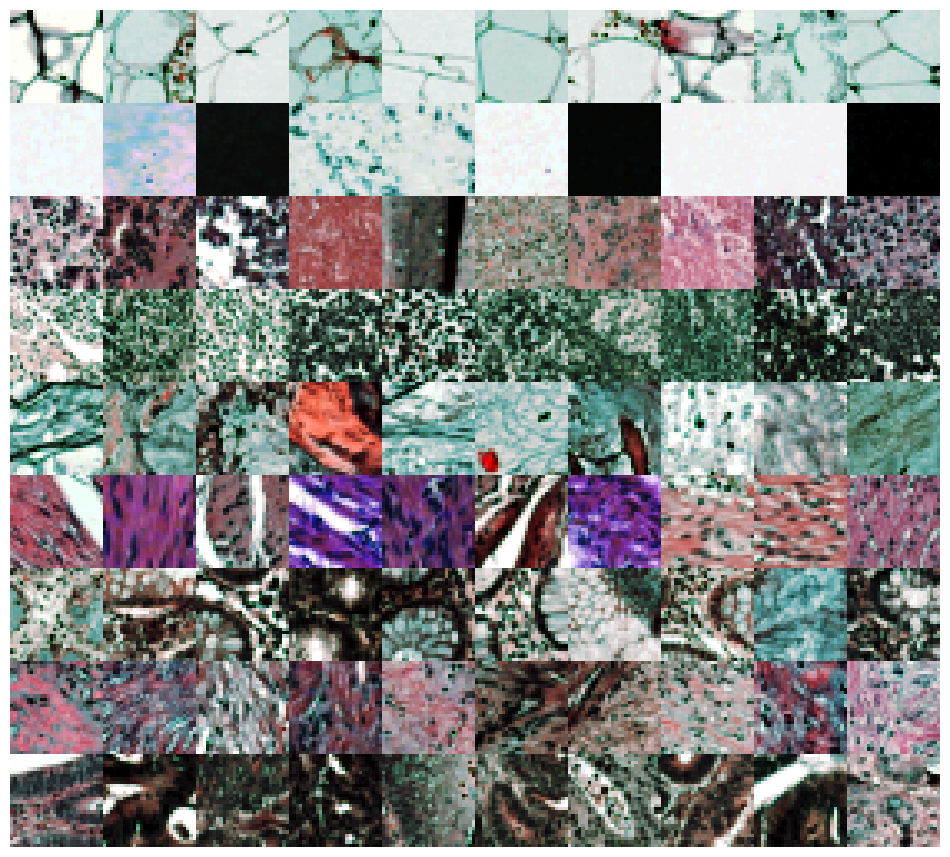
可以看到，在放大对比度之后，样本显示出了更加显著的特征。

下面这张是锐化的样本图，锐化之后纹理的分界变得更加的清晰，可以帮助模型捕捉特征。

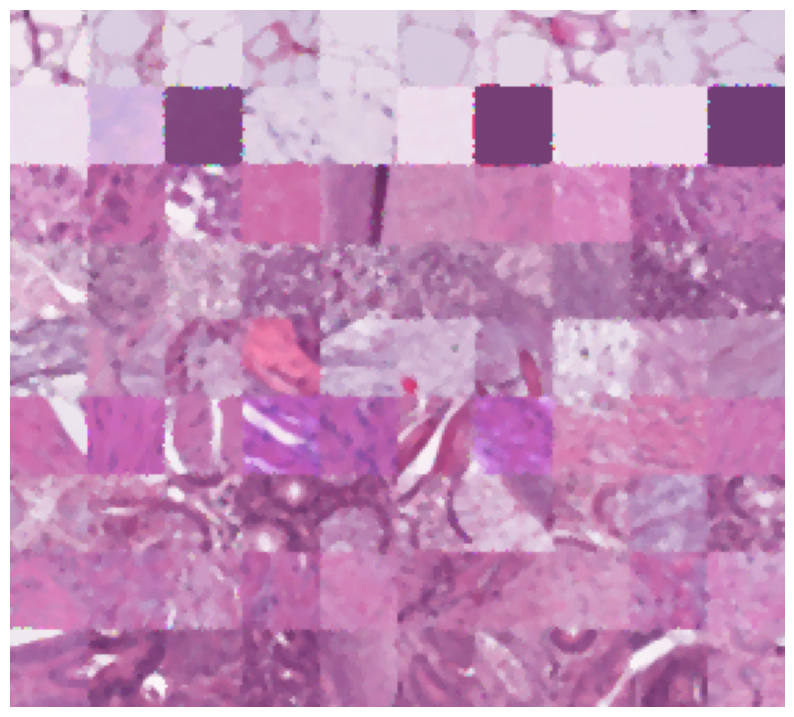


下面这张是先锐化在增强了对比度的组织图，



下图为自适应对比度+局部拉伸

同时也尝试了其他的方法来降噪，但是结果非常的模糊，可能是因为样本图片很小，降噪模糊掉了细节，这份输出结果也会尝试用于训练模型。



# Model Design

## MLP

**层级与参数详解**

1. **Flatten(input\_shape=(28,28,3))**
   * 作用：将输入的三维张量（28×28×3）“摊平”为一维向量（长度=28×28×3=2352），以供后续全连接层处理。
   * 参数：input\_shape 指定每个样本的形状（不含 batch 维度）。  
     [Keras](https://keras.io/api/layers/reshaping_layers/flatten/?utm_source=chatgpt.com)
2. **第一隐藏层 Dense(300, activation="relu")**
   * units=300：该层包含 300 个神经元（每个神经元都有自己与上一层 2352 个输入的权重向量及一个偏置）。
   * activation="relu"：ReLU 激活函数，max⁡(0,x)\max(0,x)max(0,x)，有助于加速收敛并缓解梯度消失。
   * 运算：输出 = ReLU( 输入·权重 + 偏置 )。  
     [Keras](https://keras.io/api/layers/core_layers/dense/?utm_source=chatgpt.com)
3. **Dropout(0.3)**
   * rate=0.3：每个训练步骤随机“断开”本层 30% 的神经元输出，将其置为 0，其余 70% 会按 1/(1−0.3)1/(1-0.3)1/(1−0.3) 比例缩放，保持总激活和不变。
   * 目的：防止过拟合，让网络学到更鲁棒的特征。  
     [Keras](https://keras.io/api/layers/regularization_layers/dropout/?utm_source=chatgpt.com)
4. **第二隐藏层 Dense(300, activation="relu") + Dropout(0.2)**
   * 同第一隐藏层，但这里 rate=0.2，保留更多神经元（80%），作为一种正则化手段。
   * 保持与前层相同规模（300 个神经元），有利于网络学习更深的非线性组合。
5. **第三隐藏层 Dense(300, activation="tanh") + Dropout(0.2)**
   * activation="tanh"：双曲正切函数，输出范围 [−1,1][-1,1][−1,1]，在某些任务上能提供对称激活特性。
   * 通常 tanh 收敛比 ReLU 慢一些，这里可视为 “特征变换层”，为输出层提供多样化表示。
   * 同样用 Dropout(0.2) 做正则化。
6. **输出层 Dense(9, activation="softmax")**
   * units=9：9 个神经元，对应 9 类别的打分。
   * activation="softmax"：对打分做指数归一化，输出各类别的预测概率，和为 1。

**compile 配置**

* **optimizer=Adam(learning\_rate=1e-3)**
  + Adam 优化器：结合了动量（第一矩矩估计）和自适应学习率（第二矩估计），在大多数任务上收敛稳定而高效。
  + learning\_rate=0.001：初始步长，若过大易振荡，若过小收敛慢；1e-3 是常见默认值。 [Keras](https://keras.io/api/optimizers/adam/?utm_source=chatgpt.com)
* **loss="sparse\_categorical\_crossentropy"**
  + 用于整数标签（0–8）的多分类交叉熵损失。
  + 内部会先把网络输出概率与 one-hot 编码标签做交叉熵计算，再求均值。
* **metrics=["accuracy"]**
  + 在训练/验证/测试过程中，会额外计算并报告分类准确率。

**这种设计的优点**

1. **结构简单易调**：Sequential + Dense + Dropout，适合初步对比不同预处理或超参下 MLP 表现。
2. **ReLU + tanh 组合**：前两层用 ReLU 快速提取稀疏高维特征，第三层 tanh 提供对称非线性，有助于平滑边界。
3. **多级 Dropout**：0.3 → 0.2 → 0.2 的正则化，可逐层防止过拟合，同时保留足够容量学习复杂模式。
4. **Adam 优化器**：适合多参数网络且对超参数不太敏感，收敛快。

调用

**数据归一化**

**作用**：将像素值从原本的 0–255 缩放到 0–1 区间，

 **好处**：

1. 加快神经网络训练收敛速度（避免梯度爆炸／消失）；
2. 让所有输入特征落在相同量级，提升优化器效率。

**模型训练（fit）**

* **validation\_split=0.1**
  + 自动从训练集中拿出最后 10% 作为验证集，用于监控过拟合与早停。
* **epochs=20**
  + 最多完整遍历训练集 20 次；
  + EarlyStopping 触发时会提前终止。
* **batch\_size=64**
  + 每次梯度更新用 64 张图，权衡训练速度与内存占用。
* **EarlyStopping 回调**
  + **监控指标**：val\_loss（验证集损失），而非训练损失；
  + **patience=5**：若 5 个 Epoch 内验证损失都没变好，就认为模型已收敛／开始过拟合，停止训练；
  + **restore\_best\_weights=True**：训练结束后自动回滚到验证集上表现最好的那次权重。
* **优点**：
  + 节省不必要的训练时间；
  + 有效避免过拟合，提升最终模型在新数据上的泛化能力；
  + 自动“找”到最优 Epoch，无需手动对比多个模型文件。

**绘制损失曲线**

* 作用：把每个 Epoch 的训练损失和验证损失画在一张图上，
* 优点：
  + 可视化训练过程，直观判断是否收敛、是否过拟合（验证损失开始上升时即过拟合）；
  + 有助于调参，例如增大或减小 learning rate、调整网络结构。

**测试集评估**

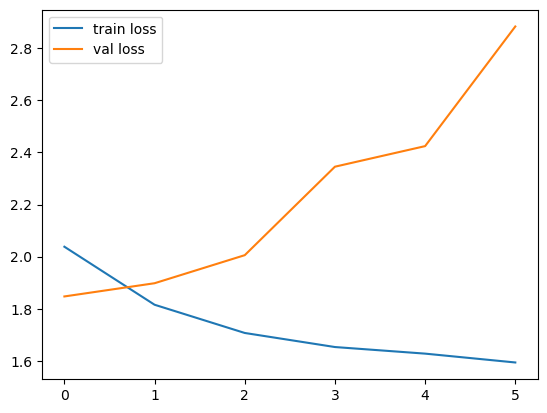
**** evaluate：在完全独立的测试集上计算损失和指标（accuracy）。

 优点：

* 给出模型在未见过数据上的真实表现；
* 与训练/验证阶段区分，防止信息泄漏。

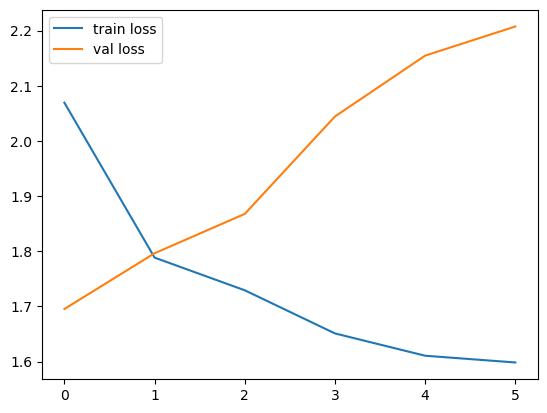
## 模型设计观察结果

* 1. 单纯的增加层数并不能直接带来准确性的提升。
  2. 训练过程中发现模型对训练数据的过度拟合依旧是主要问题，在初始的模型中，已经通过设置dropout比例，设置早停来防止过度拟合，但是问题依然严重。



尝试在通过在训练数据中认为的制造噪音来提高模型的泛化能力。

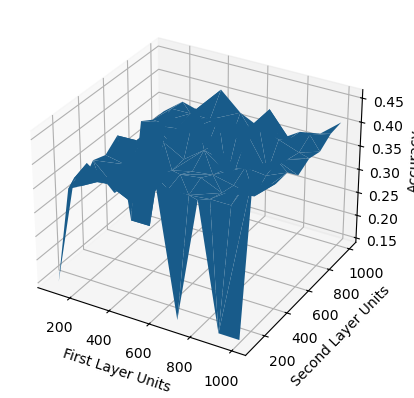
在人为的添加翻转，旋转，缩放的噪音数据进行数据增强后，模型的泛化能力得到显著的提高，随着与训练数据的拟合，验证数据的损失上升程度得到了控制。



## 调整模型深度和神经节数量对训练的影响：

模型的训练已对所有随机因素设置了种子，确保相同的模型每次fit的获得相同的结果

下述实验基于未经过预处理的原始训练数据进行。



图片描述了一二层神经节数量和测试数据集精确度之间的关系，因为算力原因只探索了两层神经网络。从上图中可以看到，神经节的数量和准确性之间并不存在明显的相关关系精确度最高的组合是400 1000，当然这和随机种子也有关系。