# 实现报告

图示, 示意图

描述已自动生成

[实现报告 1](#_Toc104647582)

[使用到的组件： 2](#_Toc104647583)

[在MATLAB中：训练神经网络分类器 2](#_Toc104647584)

[读取图像 3](#_Toc104647585)

[分类图像并决定压缩程度 4](#_Toc104647586)

[提取图像YCbCr通道 4](#_Toc104647587)

[图像区块分割与合并 5](#_Toc104647588)

[区块的2D-DCT压缩 6](#_Toc104647589)

[合并图像通道并输出 7](#_Toc104647590)

[计算损失 7](#_Toc104647591)

## 使用到的组件：

* MATLAB R2022a
* Simulink
* DSP System Toolbox
* Image Processing Toolbox
* Computer Vision Toolbox
* Deep Learning Toolbox
* Deep Learning Toolbox Model for ResNet-50 Network(仅在训练时需要)
* Parallel Computing Toolbox, GPU Coder(可选，为了GPU加速训练)

## 在MATLAB中：训练神经网络分类器

实现程序位于MATLAB\_Classifier\_Training\Classifier.m

* 从[该网站](https://doi.org/10.6084/m9.figshare.4595323.v1)下载图像训练集，解压于images文件夹。训练集有两个子文件夹，indoor代表室内照片，outdoor代表室外照片，每个类别均有400张。将每张照片拉伸并缩放到224x224分辨率，存于images\_resized224文件夹，保留目录结构。
* 使用MATLAB中的imageDatastore()函数读入训练集，该函数可根据子文件夹名添加图像标签。
* 使用MATLAB的“深度网络设计器”功能导入ResNet-50神经网络，并将代码导出到getLayers.m函数中。修改第一层输入层和倒数第二层全连接层的大小分别为224x224x3和2，以匹配输入图像分辨率和输出的两种类型(室内和室外)。
* 尽管可以直接使用MATLAB内的trainNetwork()函数来训练网络并保存，但为了能方便地在Simulink中导入模型，我直接使用“深度网络设计器” 功能来训练网络，如图表

  描述已自动生成图所示：之后选择将模型导出到Simulink，即可获得可在Simulink中使用的图像分类模型：trainedNetwork\_224x224.mat。

## 读取图像

图示

描述已自动生成

使用Image From File模块读入图片，并立即显示以方便后续图像对比。

## 分类图像并决定压缩程度

图示

描述已自动生成

由于训练的深度神经网络的输入大小为[224x224x3](#_在MATLAB中：训练神经网络分类器)，因此先将输入的图像缩放到224x224分辨率，再传入神经网络予以分类。神经网络输出一个1x2的向量，第一个值表示该图像为室内照片的概率，第二个值表示该图像为室外照片的概率，两个值的和为1。在上图的示例中，输入图像为我的寝室内的照片，因此认为照片为室内照片的概率为99.43%，可认为已被准确分类。

之后将分类结果传入MATLAB函数CompressV，该函数在图像分类为室内时输出压缩系数12，分类为室外时输出6，并传输到之后的DCT后压缩使用的掩码生成器中，以生成对应大小的掩码。值得注意的是压缩系数在传入到Cb、Cr两个色度分量时被减去了2，这是因为人眼对明度敏感性不如色度，因此适当减少色度信息有助于更高效地压缩。

## 图示 描述已自动生成提取图像YCbCr通道

先将图像由int8类型转换到double以便后续处理，之后使用MATLAB函数SepYCbCr将RGB格式转换成YCbCr格式，并分通道输出。

## 图像区块分割与合并

图示

描述已自动生成

为了对整个图像进行分块DCT，该实验使用了Computer Vision Toolbox中的vipmisc库(MATLAB\R2022a\toolbox\vision\vision\vipmisc.slx)里的分块与拼接。该库提供了Input和Output一对模块，可以用自定义的区块大小(本实验中为8x8)分割图像并在处理后拼接。在上图所示子系统中，为了遍历整个图像，该子系统运行了区块个数次。

## 区块的2D-DCT压缩

图示

描述已自动生成

形状

低可信度描述已自动生成在该子系统中，输入的是每个8x8的区块。在In1输入区块图像后，对其应用DCT，使得图像的高频信息靠近右下角，低频信息靠近左上角。

压缩系数=10对应的掩码

端口P传入了[由分类器和通道类型决定的压缩系数](#_分类图像并决定压缩程度)，根据该系数使用MATLAB函数Mask生成掩码。根据压缩系数，决定了有多少个右上-左下对角行填1，而其余的填0，如右图所示；将该掩码与应用DCT后的区块相乘，使得区块中只保留了靠近左上填1的低频部分，靠近右下填0的高频部分则被舍弃。据此，实现了根据压缩系数决定了保留多少图像高频信息。

对处理后的图像应用DCT逆变换，得到了压缩高频信息后的8x8区块，并如[上述](#_图像区块分割与合并)拼接为完整图像。

## 图示 描述已自动生成合并图像通道并输出

经过上述处理后得到的是处理后的[YCbCr通道](#_提取图像YCbCr通道)，因此MATLAB函数Combine将图像的Y、Cb、Cr三个通道进行了合并，并转化成RGB格式后输出图像。该图像即为经过DCT与压缩处理后的完整图像，本实验流程到此结束。

## 图示 描述已自动生成计算损失

在MATLAB函数Eval中使用psnr()方法来计算处理后的图像与原图相比的峰值信噪比(Peak signal-to-noise ratio)。使用Display模块监听PSNR输出，该值越大，代表输出图像与原图像相比损失较少；该值越小，代表输出图像与原图像相比损失较大。­