

基于带有颜色的纹理描述符的瓷砖表面色差检测技术研究

摘要：

关键词：瓷砖；色差检测；纹理识别；深度学习；

优化点：

- 难样本（那些尽管分类正确，但是 softmax 中概率显示并不高，可能只比第二高一点），可视化看为什么会出现这样的情况，具体它关注的是哪里
- 问题描述，我的研究方案可以解决那些尽管纹理不同，但是属于同一类的瓷砖的色差问题，如果用传统的方法，只用颜色去比较，因为纹理不同，那么颜色直方图中落入各个 bin 中的像素点的数量可能会差别很大，纹理不同其实就直接导致了色差，而我们现在解决的方案可以解决这个问题！！因此需要引入纹理。
- 实验结果（图片），在验证好之后可以将实验结果展示出来，说明那些尽管纹理不同但是属于同一类的那些瓷砖，不会因为纹理不同而产生色差问题。
- 复现一些模型（尽量把有代码的那些模型跑一下，在对比实验里填上去）

1 引言（过于口语化，需要改）

瓷砖表面的色差是判定瓷砖成品质量的重要指标之一。在瓷砖实际生产过程中，原料、坯料加工，成形和烧成等多个环节，控制稍有偏差，都有可能导致色差^[1]，而色差会极大影响最终瓷砖的装饰效果。因此，色差检测是瓷砖出厂前必经的工序。目前瓷砖色差依赖人工分拣，这种传统的检测方法存在诸如易受到外界环境的干扰、容易出现误判和效率低等问题。并且国外较成熟的瓷砖检测系统无法在国内得到实际应用^[3]。

为解决人工目测带来的问题，在上个世纪九十年代，开始出现利用视觉检测技术实现自动化色差检测的方法，其核心在于如何提取能够精确表征图像的颜色描述符。如提取 RGB 或 CIELAB 颜色空间图像的各通道颜色均值作为颜色特征以达到颜色分级的效果^[2-3]；利用 K-means 聚类方法进行图像分割，从而将多色砖的色差检测转换成多个单色砖的色差检测^[4]。此类方法仅适用于单色且纹理简单的瓷砖，而对颜色和纹理日益多元化的瓷砖无能为力。此后出现了大量比较图像颜色直方图分布相似性的方法^[5-9]，其中不乏有方法对图像进行预处

理^[6]，使图像符合人类视觉系统对颜色的感知特性。这些方法存在的一个主要问题是，它们仅关注了颜色的比例分布，而忽略了颜色的空间分布，因此，这类方法同样无法检测纹理复杂的瓷砖。为了解决以上问题，多项研究将纹理的概念引入到图像表征中来^[10-14]，[这里英文文献还可以加，从第10篇找](#)，希望利用纹理来近似颜色的空间分布，其中利用二阶统计量（灰度共生矩阵）和多尺度空间能量分布（小波变换）来表征图像纹理为主流方法，基于纹理与颜色描述符相结合的方法大幅度提升了瓷砖色差检测的性能。尽管如此，对于纹理与颜色均复杂的瓷砖的色差检测任务依然面临着两个具有挑战性的问题（1）如图1所示（[图中显示直方图，可视化差异](#)），相同类别的瓷砖存在不同的纹理。传统的检测手段将纹理的差异反映成色差，从而产生误检。虽然纹理的引入一定程度上缓解了误检的发生，但由于目前纹理特征的表达过于[简单](#)，还是无法使色差检测达到一个理想的性能。（2）人类对于不同颜色的色差感知不同，因此如何选取合适的颜色描述符来表征图像很大程度上影响着最终色差检测的结果，而目前仅能通过网格搜索的方式，尝试比较各种手工提取的颜色描述符和纹理描述符的性能的好坏，耗时且很难达到最优。

近年来，深度学习飞速发展，基于深度学习的算法在计算机视觉的各个领域都得到了很好的应用，如缺陷检测^[...]、色差校正^[...]。而对瓷砖色差检测领域相关的研究仅限于对分类器的讨论。在本文中，鉴于深度学习强大的特征学习和表示能力，以及瓷砖表面色差检测存在的问题，我们提出了一个新颖且有效的色差检测框架。我们的方法基于这样的观察：在图像分类^[下面的14, 15]（包括纹理识别）、分割^[16]等领域，深度学习提取的特征通常优于手工提取的特征，且通过实验发现，基于CNN提取的纹理特征对于颜色的变化是不敏感的，即纹理特征中几乎不包含有效的颜色信息。

具体而言，该框架包括用于提取纹理与部分颜色信息的多任务模块以及色差检测模块。基于深度学习的多任务模块首先通过纹理识别分支与颜色分支提取完整的纹理描述符与部分颜色特征。其次色差检测模块将这两部分特征与基于颜色直方图的颜色信息相结合。最后，通过多层感知机网络完成色差检测。该方法在两个方面不同于现有的工作：纹理特征的提取与颜色特征的表达。

我们的贡献总结如下：

（1）与现有基于纹理与颜色相结合的方法（比如[10-14]）不同，这些方法通过基于统计或是滤波的方式提取纹理特征，我们引入基于深度学习的多任务框架提取更有效的纹理特征。这不仅允许我们提取准确的纹理描述符，且额外获得的颜色特征与基于颜色直方图的颜色信息相结合，使得对于图像的颜色描述更为全面。

(2) 我们首次在多任务框架中引入纹理识别与颜色分类任务。在纹理识别网络中通过引入纹理增强模块 (TEM)、多方向-金字塔纹理特征提取模块 (MD-PTFEM) 以及直方图层 (HL)，并使用紧凑双线性池化作为特征聚合的方法，在几个基准数据集上实现了具有竞争力的性能。通过设计全局注意力层 (GAL)，颜色分类分支同样获得了理想的分类精度。(改写法，表述解决了第二个挑战)

(3) 得益于我们提出的色差检测框架的有效性，在与几种常用的色差检测方法的比较中都获得了 SOTA 的效果。(改写法，表述解决了第一个挑战)

2 相关工作

2.1 瓷砖表面色差检测 (国内外先都写着)

2.2 纹理识别

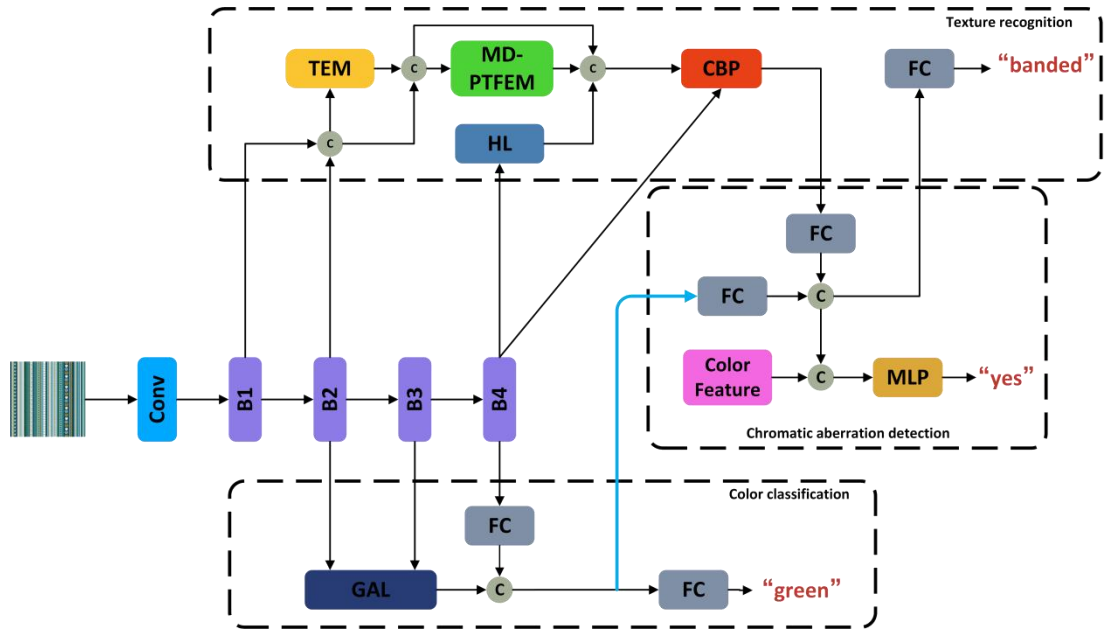
纹理识别在计算机视觉领域有着广泛的应用，

2.3 多任务学习

2.4 双线性池化

3 方法

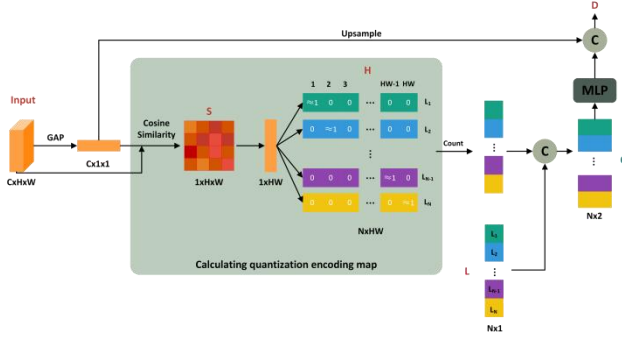
在本节中，我们详细介绍了提出的瓷砖表面色差检测方法。首先我们在 3.1 小节中描述了色差检测网络的总体架构。其次在 3.2 小节中阐述了纹理识别网络中的纹理增强模块 (TEM)、多方向-金字塔纹理特征提取模块 (MD-PTFEM)、直方图层 (HL) 和紧凑双线性层 (CBP)。然后在 3.3 小节中介绍了颜色分类网络，包括了全局注意力层 (GAL)。接着我们在 3.4 节中介绍了用于平衡纹理识别任务和颜色分类任务之间学习的损失函数。最后我们详细介绍了色差检测的过程。



3.1 总体架构

色差检测网络总体的架构如图 1 所示。色差检测的过程可以分为两个步骤，第一步是提取图像带颜色的纹理特征。这一步我们设计了一个硬参数共享的多任务网络，该网络有三部分组成：基础网络、纹理识别分支和颜色分类分支。对于基础网络，我们使用 ResNet50[14]。对于纹理识别分支，我们将基础网络中第一层和第二层输出的特征进行下采样至基础网络输出大小，然后通过串联融合，依次执行纹理增强模块和多方向-金字塔纹理特征提取模块，同时我们利用直方图层对基础网络第四层的输出进行了统计，最终将这两个输出进行拼接，与基础网络第四层输出的结构纹理信息采用双线性池化的方式融合，为了不引入额外过多的参数，我们选择了紧凑双线性池化^[6]。对于颜色分类分支，我们引入了全局注意力层，首先将基础网络的第三层和第二层输出输入全局注意力层，再与最后一层的输出进行拼接获得最终的特征图。当然，为了让颜色的信息嵌入到纹理特征中，我们在纹理识别前，将颜色分类网络的最后一个全连接层的输入与纹理识别网络紧凑双线性池化后的输出做了有效的融合。

第二步是将纹理特征与颜色特征拼接作为图像的描述符，输入多层感知机网络判别色差。在 CIELAB 和 RGB 颜色空间中提取瓷砖表面图像颜色特征，该颜色特征由 L、a、b 三通道的颜色直方图以及 R、G、B 三通道的均值和方差构成。将纹理和颜色特征向量进行拼接，输入到预先训练好的多层感知机网络中，即可判别瓷砖是否存在色差。



3.2 纹理识别分支

为了能够充分利用低层的纹理特征，提出了纹理增强模块（TEM）和金字塔纹理特征提取模块（PTFEM）^[7]。这两个模块最初是用在语义分割领域，我们将其首次应用于纹理识别领域，并对金字塔纹理特征提取模块进行了改进，将其命名为多方向-金字塔纹理特征提取模块（MD-PTFEM）。

纹理增强模块^[7] 从主干网络中提取的低层特征，尽管保留一定的空间维度信息，但是依然会都丢失很多纹理细节，使得图像的对比度下降，因此该模块就是受到了直方图均衡化的启发，增强图像的纹理细节，使得我们能够在下一个模块更好的提取纹理特征。

如图 2 所示，我们首先提取图像中纹理的统计特征。我们将输入特征图 $A \in \mathbb{R}^{C \times H \times W}$ 进行全局平均池化得到全局平均特征 $g \in \mathbb{R}^{C \times 1 \times 1}$ ，计算 A 中每个空间位置 $A_{i,j} (i \in [1, H], j \in [1, W])$ 与 g 的余弦相似度得到 $S \in \mathbb{R}^{1 \times H \times W}$ 。然后重新调整 S 的维度至 $\mathbb{R}^{1 \times HW}$ ，将 S 量化成 N 个等级，得到量化等级集合 $L = \{L_1, L_2, \dots, L_N\}$ ，这 N 个等级的值是通过在 S 的最小值和最大值之间等分 N 个点获得的。接着计算 S 中的每一个空间位置 $S_i (i \in [1, HW])$ 归属于哪一个等级，最终 S_i 会得到一个类似于独热编码的量化编码向量 $H_i \in \mathbb{R}^N (i \in [1, HW])$ ，在 H_i 中，只有 S_i 所属那个等级的维度才不为 0。与独热编码不同， H_i 中的值并非只有 0 和 1，而是 0 和逼近 1 的值，这种方式更加平滑，可以避免梯度消失的问题。根据 H 对每一个量化等级进行统计，并做归一化操作，结果与量化等级进行拼接，得到 $C \in \mathbb{R}^{2 \times N}$ ，其中第一维表示每个量化级别，第二维表示相应的统计量。由于 C 是 A 的相对统计信息，为了进一步获得绝对统计信息，我们将 g 上采样至 $\mathbb{R}^{C \times N}$ ，并且与经过 MLP 的 C 进行拼接，得到最终的输出 $D \in \mathbb{R}^{C_1 \times N}$ ，也就是每一个量化等级的统计特征都可以由一个 \mathbb{R}^{C_1} 的向量来表示。其中 MLP 是为了升高 C 的维度，它由一个卷积层和 Leaky ReLU 层组成。

了 $A_{i+s,j+t}$ 中各个空间位置所属的量化等级。 $\hat{H} \in \mathbb{R}^{N \times N \times H \times W}$ 更加细致全面的统计了相邻像素之间的量化相关性。

根据 \hat{H} 对每一对可能的量化共线进行统计，并做归一化操作，结果与所有量化共线 $\hat{L} = \{L_{m,n} = \{L_m, L_n\} | m, n \in [1, N]\}$ 进行拼接，得到 $C \in \mathbb{R}^{3 \times N \times N}$ ，其中前两维表示每对量化共线，第三维表示相应的统计量。最后与纹理增强模块一样，我们将平均特征上采样至 $\mathbb{R}^{C \times N \times N}$ ，并且与经过 MLP 升维的 C 进行拼接，得到绝对量化共线统计特征 $D \in \mathbb{R}^{C_1 \times N \times N}$ 。

基于灰度共生矩阵（GLCM）的思想，使用统计描述符来描述纹理信息。由于深度卷积神经网络有强大的特征提取能力，我们让网络自动学习类似于对比度、逆方差等有用的统计信息。将量化共线的统计特征 D 经过一个 MLP 层得到 $F \in \mathbb{R}^{C_2 \times N \times N}$ ，最后通过公式（3）进行逐层平均获取纹理特征 T

$$T = \frac{\sum_{m=1}^N \sum_{n=1}^N F_{:,m,n}}{N \cdot N} \quad (3)$$

金字塔结构 瓷砖图像上的纹理复杂多变，不同区域包含的信息不尽相同，提取不同尺度的纹理特征有利于提高最终纹理特征的判别性，从而提高纹理识别性能。因此，采用金字塔结构从多尺度特征中提取纹理信息，整个过程如图 4 所示。首先我们将特征图分别输入到四个不同尺度的区域划分分支，在此处特征图将被分为不同数量的子区域。每一个子区域都会经过多方向-纹理特征提取模块（MD-CFEM）以获得该区域对应的纹理特征。**然后分别将四个分支输出的纹理特征图通过双线性插值上采样到输入特征图的大小，并将它们拼接起来，得到最终的输出。**

金字塔结构与多方向-纹理特征提取模块的结合，我们将其命名为多方向-金字塔纹理特征提取模块（MD-PTFEM）。

直方图层 神经网络在深层提取到的特征拥有更大的感受野，因此它们所表达的含义更为全局，更加偏向于特定领域。包含丰富语义信息的深层特征是我们能够进行精准分类的关键，所以我们利用直方图层描述深层特征的局部空间分布。直方图层的实现与 Peebles、Xu 和 Zare^[8] 提出的直方图层一样。利用 Radial basis functions （RBFs）建模直方图，将 RBF 的平均值（ μ_{bk} ）作为 bin 中心，带宽（ γ_{bk} ）作为 bin 宽度，在 $S \times T$ 局部空间内计算归一化频率计数 Y_{rcbk} ，对于输入 x 的第 k 个通道的直方图值（特征）的分箱操作定义为：

$$Y_{rcbk} = \frac{1}{ST} \sum_{s=1}^S \sum_{t=1}^T e^{-\gamma_{bk}^2 (x_{r+s,c+t,k} - \mu_{bk})^2} \quad (4)$$

其中， b 表示直方图的第 b 个 bin， k 表示输入 x 的第 k 个通道， r 和 c 表示直方图特征图上的空间位置。若直方图 bin 的数量为 B 个，对于输入 x 的每一个通道都会计算 B 张直方图特征图，每一张直方图特征图的 bin 中心和宽度都是不同的，其中特征图上的每一个空间位置的特征值，反映了它离当前 bin 中心靠近的程度，从公式 (4) 中可以看出，特征值越靠近 bin 中心，越接近 1，反之，则越接近 0。与标准的直方图操作相比，这种软分箱的操作对于 bin 中心和宽度的小变化具有鲁棒性，因为它允许存在一定的误差。

具体的实现：首先将主干网络第四层的输出作为输入，使用 $1 \times 1 \times K$ 的卷积进行降维，其中， K 表示新特征图的通道数。然后对每一张特征图应用 $1 \times 1 \times B$ 的卷积，计算每一个特征与各个 bin 中心的距离（公式 (4) 中的减去 μ_{bk} ）。卷积核中的 bias 作为可学习的 bin 中心，权重固定为 1。接着对每一张特征图应用 $1 \times 1 \times B$ 的卷积，将中心特征乘以 bin 宽度（公式 (4) 中的乘以 γ_{bk} ），以合并 bin 的特征分布。卷积核的权重作为可学习的 bin 宽度，bias 固定为 0。根据公式 (4) 中的 RBF 函数进行平方、求反，应用指数操作，计算每一个特征对各个 bin 的贡献，该贡献的值被限制在 0 和 1 之间。最后利用平均池化（ $S \times T$ 大小的滑动窗口）计算局部空间区域中特征的贡献，从而得到各个 bin 的特征的归一化频率计数。

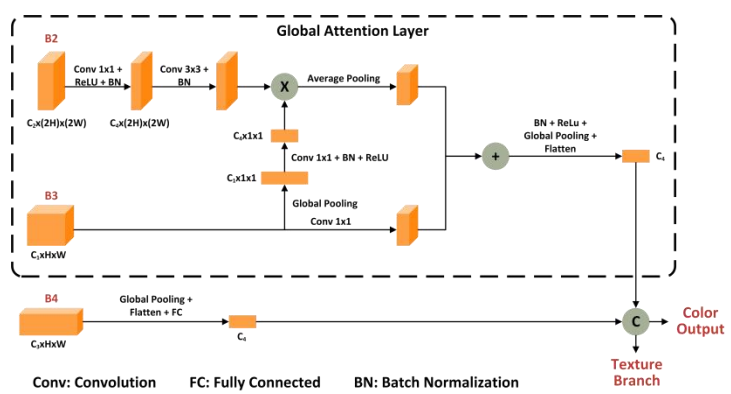
紧凑双线性池化 用 $X \in R^{C \times H \times W}$ 表示 CNN 中某一特定层输出的特征图，其中 H ， W 和 C 分别表示特征图的高度、宽度和特征通道数。 $x_i \in R^C (i \in [1, HW])$ 表示在特征图 X 上第 i 个空间位置的特征向量。对于每一个空间位置上的特征，计算其外积 $x_i x_i^T$ ，将得到的所有矩阵相加，最终得到 $I \in R^{C \times C}$ 的双线性特征。我们将 X 重新调整维度至 $R^{C \times HW}$ ，它的双线性特征可以定义为：

$$I = \sum_{i=1}^{HW} x_i x_i^T \quad (5)$$

可以看出双线性池化提取了特征的二阶信息。在图像分类中，二阶特征往往比一阶特征含有更多的信息，且更具判别性。但双线性池化存在一个明显的缺点：计算复杂度高。双线性池化最终得到的输出维度是平方级别的，例如 $C = 512$ 时，双线性特征 I 的维度为 256K，这不仅对计算成本还是存储成本都是昂贵的。为了捕获纹理细节的统计信息与空间结构信息（主干网络最后一层的输出）的成对相关性，以最大限度的发挥这两个互补信息的优势，我

们采用紧凑双线性池化[6]的方式联合这两个互补信息。Gao 等人基于 Random MacLaurin [9] 和 Tensor Sketch[10]提出了两种近似方法。这两个方法都是通过大的随机矩阵将局部特征投影到低维，再用低维特征的 Hadamard 积近似双线性池化特征。我们采用[6]中基于 Tensor Sketch 的方法实现紧凑双线性层。~~（要不要加上 Tensor Sketch 算法的图待定）~~

3.3 颜色分类分支



我们使用正常的 DTD^[11]数据集与灰度化的 DTD 数据集分别进行纹理分类的实验，结果显示，纹理识别的精度几乎没有差别，说明了纹理识别网络最终提取的纹理特征对于颜色的变化是不敏感的，即，纹理特征中的颜色信息几乎为零。鉴于深度卷积神经网络强大的提取能力，我们将通过颜色分类分支提取瓷砖图像的颜色特征，作为色差检测时颜色特征的补充信息，以更准确的描述图像的颜色。

全局注意力层 我们通过实验发现，对于带有纹理的图像，在特征提取的过程中，**包含最丰富颜色信息且能最准确反映图像颜色特征的位置**，并非网络的最后一层，而是前一层。因此，我们提出的全局注意力层利用的是主干网络第二层和第三层输出的特征。我们通过具有丰富颜色信息的高层特征加权低层特征以精确的选择低层特征中的颜色信息。具体而言，我们对高层特征进行全局平均池化，该池化特征作为全局上下文指导低层特征选择颜色信息。实现步骤如图 5 所示。我们将主干网络第三层的输出进行全局平均池化并依次执行 1x1 的卷积，批归一化和 ReLU 非线性激活，从而得到全局上下文。接着对第二层的输出进行降维，然后与全局上下文相乘。最后将加权的第二层特征与第三层特征相加并与第四层的特征融合，得到描述颜色的图像特征。该颜色特征有两个作用。其一是作为颜色分类的特征。第二个是与纹理识别网络中提取的纹理描述符融合得到带有颜色的纹理描述符。

3.4 多任务损失函数

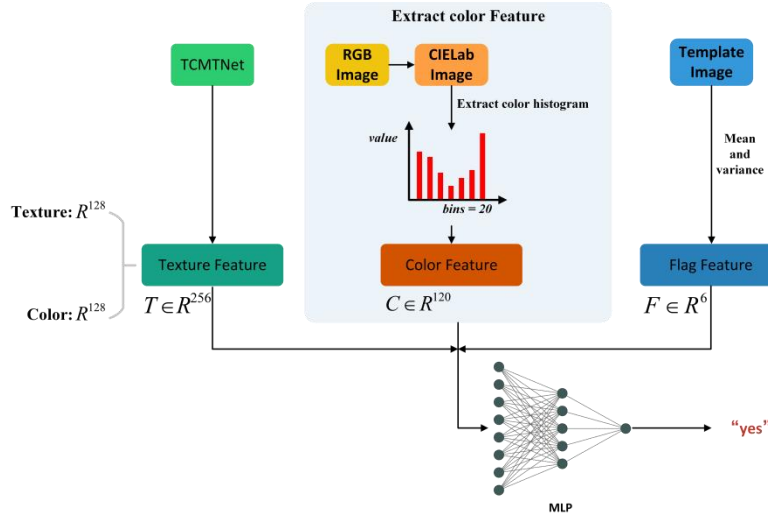
我们的多任务网络仅涉及到两个任务：纹理识别任务和颜色分类任务。我们使用最简单的加权线性融合的方式来表示总体损失，具体来说，我们通过如下公式来联合训练纹理与颜

色任务：

$$L_{total} = w_t L_{texture} + w_c L_{color}, \quad (6)$$

其中， $L_{texture}$ 和 L_{color} 分别表示纹理识别任务和颜色分类任务的损失， w_t 和 w_c 分别表示纹理识别任务和颜色分类任务损失的权重系数，用于平衡两个任务的训练。 w_t 和 w_c 满足 $w_t + w_c = 1$ 的条件。在我们的实验中， w_t 和 w_c 分别设置为 0.88 和 0.12。对于 $L_{texture}$ 和 L_{color} ，我们均使用了普通的交叉熵损失。

3.5 色差检测



色差检测的总体流程如图 6 所示。用于描述图像的特征分为三部分。第一部分为多任务网络输出的带有颜色信息的纹理描述符 $T \in R^{256}$ ，其中包括从纹理识别网络中输出的 128 维的纹理特征和颜色分类网络中输出的 128 维的颜色特征。

第二部分的特征是手工提取的图像颜色描述符 $C \in R^{120}$ 。首先我们将待测瓷砖图像 $I \in R^{3 \times H \times W}$ 从 RGB 颜色空间转换到更加符合人类视觉感知的 CIE L*a*b* 颜色空间，得到 $I' \in R^{3 \times H \times W}$ ，分别提取 L^* 、 a^* 和 b^* 通道的颜色直方图，通过 bin 中心以及各个 bin 对应的统计量来表征颜色直方图。 $C = (H_{L^*}, H_{a^*}, H_{b^*})$ ，其中， $H_i = (B_i, S_i)$ ， $i = L^*, a^*, b^*$ ， H_i 表示 i 通道的颜色直方图特征， B_i 表示由 i 通道颜色直方图的 bin 中心构成的特征向量， S_i 表示由 i 通道颜色直方图中各个 bin 对应的统计量构成的特征向量。取图像 I' 中 i 通道像素的最大值和最小值，将其等分 N 个点得到 $B_i = (b_1^i, b_2^i, \dots, b_N^i)$ ：

$$b_n^i = \frac{\max(i) - \min(i)}{N} \cdot (n - \frac{1}{2}) + \min(i), \quad n \in [1, N] \quad (7)$$

对于图像 I' 中 i 通道的每一个空间位置 $I'_{h,w}^i (h \in [1, H], w \in [1, W])$ ，其属于第 n 个 bin 需满足如下条件：

$$-\frac{\max(i) - \min(i)}{2 \times N} \leq I'_{h,w}^i - b_n^i \leq \frac{\max(i) - \min(i)}{2 \times N} \quad (8)$$

将图像 I' 中 i 通道的所有像素分配到各个 bin 中并统计得到 $S_i = (s_1^i, s_2^i, \dots, s_N^i)$ ， s_n^i 为第 n 个 bin 包含的像素总数。最终我们会将 B_i 与 S_i 做归一化操作，具体而言，由于 $L^* \in [0, 100]$ ， $a^*, b^* \in [-128, 127]$ ，我们对各个通道的 B 分别做归一化的操作， $B_{L^*} = B_L / 100.0$ ， $B_{a^*} = (B_{a^*} + 128) / 255.0$ ， $B_{b^*} = (B_{b^*} + 128) / 255.0$ 。对于 S_i ，我们对其做 L1 归一化操作： $S_i = S_i / (H \times W)$ 。

第三部分的特征 $F \in R^6$ ，被称为标记特征，它被用来处理一些特殊情况。这部分特征并非通过待测瓷砖 I 中提取，而是 I 的模板图像 G 。因为同一类的瓷砖，纹理的覆盖范围，纹理的位置等不尽相同，但是纹理的种类是相同的，它们都是从一块公共的模板图像上截取下来的，我们提取的就是这块模板图像 G 上的特征。在 RGB 颜色空间中，提取图像 G 三通道的均值和方差构成特征向量 F 。在色差检测中，它被用来处理纹理相同但颜色有差异的两类瓷砖可能出现的特殊情况，如有两类瓷砖纹理均相同，但颜色一深一浅，颜色较浅的瓷砖在烧制不当导致颜色加深，出现色差，该颜色与颜色较深那类的瓷砖相似，此时，仅仅根据前两部分特征进行色差检测，则会认定该块瓷砖归属于颜色深的那类，最终判定为无色差。为了避免此类特殊情况，特征 F 标记了各块待测瓷砖归属的类别，从而减少这种误判的发生。

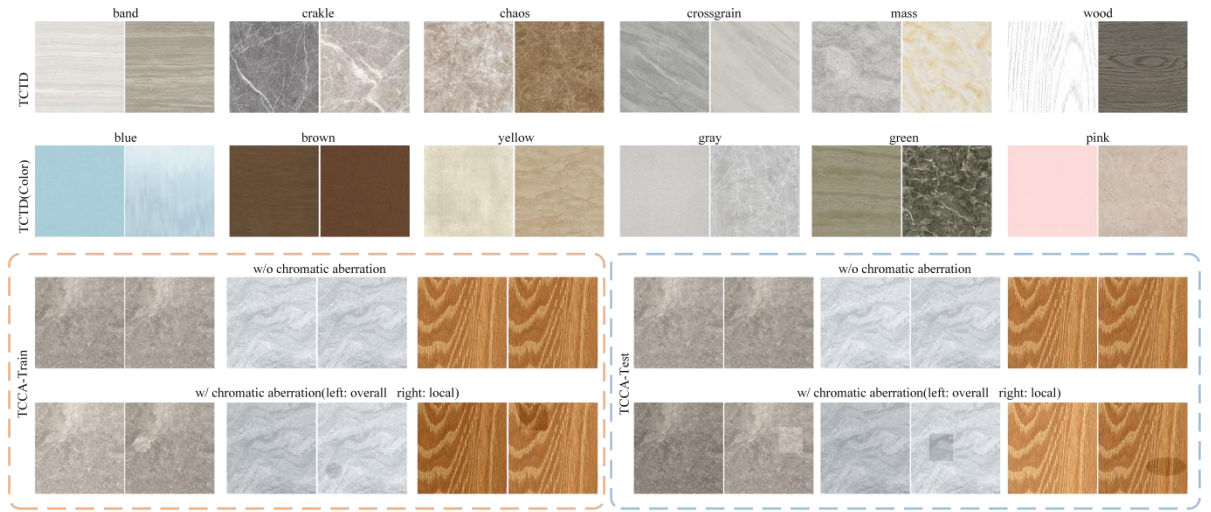
我们将这三部分特征进行通过公式 (9) 融合，得到用于描述图像的特征向量 $V \in R^{382}$ 。

$$V = \text{concat}(T, C, F), \quad (9)$$

其中， concat 表示串联操作。我们将 V 输入到仅三层的多层感知机网络，最终得到 $P \in (0, 1)$ ， P 表示特征 V 对应的那块瓷砖存在色差的概率。

$$P = \text{MLP}(V) \quad (10)$$

4 实验



4.1 数据集

我们在 3 个纹理和材料数据集上验证了纹理识别分支的有效性。**Describable Texture Dataset (DTD)** [11] 包含 47 个纹理类别，每一类 120 张图像，一共 5640 张图像。**Materials in Context 2500 (MINC-2500)** [12] 是 Materials in Context 数据集的子集，包含 23 个材料类别，每一类 2500 张图像，一共 57500 张图像。**Flickr Material Dataset (FMD)** [13] 由 10 个不同的材料类别组成，每一类 100 张图像，一共 1000 张图像。

对于颜色分类分支的实验，我们对 DTD 数据集进行了颜色标注，一共分为 14 类：黑色、黑白、蓝色、蓝绿、棕色、灰色、绿色、橙色、紫色、红色、深粉色、白色、黄色和混合色。每一类包含的图像数量分别为 278、406、288、73、625、696、447、143、121、195、254、334、708、1072。一些示例如图 7 所示。

对于色差检测的实验，我们根据佛山欧神诺陶瓷有限公司提供的瓷砖样板图制作数据集。瓷砖样板图一共 7096 张，我们从中挑选 2000 张图像，并将其分成 10 个纹理类别和 12 个颜色类别用于多任务网络的训练和测试。纹理类别：条纹、水泥、无序、裂纹、斜纹、团纹、淡纹、斑点、竖纹、木纹。每一个纹理类别 200 张图像。颜色类别：蓝色、棕色、深棕、深灰、灰色、绿色、粉色、粉灰、红棕、白色、黄色、黄棕。每一个类别包含的图像数量分别为 65、88、60、281、540、30、42、163、51、115、536、29。我们将该数据集命名为瓷砖纹理数据集 (CTTD)。在真实的生产环境中，瓷砖表面的色差往往是整体色差，局部色差很少出现。因此，我们根据专家提供的具有色差的瓷砖图像，利用 OpenCV 工具，对每一张图像执行整体或局部的颜色变换，生成 3 张色差图像，其中，一张有较明显整体色差，另一张

有较明显局部色差，最后一张有整体色差但人眼视觉感知不明显，归为无色差类别。将原图作为标准瓷砖，其余三张图像作为真实生产环境下的瓷砖，最终获得 8000 张瓷砖图像作为色差检测实验的训练集，我们将其命名为瓷砖色差训练集（CTCA-Train）。训练集包含 2 个类别：有色差和无色差。每一个类别 4000 张图像。再依照上述方法对每张图像制作 4 张不同程度色差的图像，其中两张归属无色差类别，两张归属有色差类别（一张整体色差，一张局部色差），一共 8000 张瓷砖图像作为色差检测实验的测试集，其中有色差和无色差每类 4000 张，我们将其命名为瓷砖色差测试集（CTCA-Test）。一些示例如图 7 所示。

对于 DTD 数据集，纹理识别的单任务、颜色分类的单任务以及多任务的训练和测试均使用已经提供好的训练-测试集（10 个分割子集）。MINC-2500 使用已经提供好的 5 个训练-测试集。FMD 数据集随机划分成 10 个具有推荐分割大小的训练-测试集（每一类 90 张图像用于训练，10 张图像用于测试）。TCTD 数据集根据纹理类型随机划分成 5 个训练-测试集（每一类 140 张图像用于训练，60 张图像用于测试）。我们计算所有分割集的分类精度和标准偏差，以 “ $mean \pm std\%$ ” 的形式报告结果。DTD、MINC-2500、FMD 和 TCTD 数据集上实验的结果以及色差检测的结果均基于 5 次统计。

4.2 实现细节

我们的模型在 Pytorch 上实现，并在单个 NVIDIA A30 TENSOR CORE GPU 运行。ImageNet 上预训练 ResNet18 和 ResNet50 作为多任务网络模型的主干。第一层和第二层输出的特征通过平均池化下采样至主干网络输出大小。TEM 中的量化等级设置为 128，MD-PTFEM 中的量化等级在 ResNet50 与 ResNet18 中分别设置为 8 和 12。HL 的实现与[8]相似，对于 DTD 和 FMD 数据集，直方图的 bin 个数设置为 4，对于 MINC-2500 数据集，设置为 16。我们采用[6]中提出的 Tensor Sketch 算法实现紧凑双线性池化。与全连接池化一样，我们使用了。投影的维度设置为 8192。

纹理和材料识别任务、颜色分类任务和多任务网络的训练方式均相同，且与[8]一致。DTD、MINC-2500 和 FMD 数据集中所有的图像均重新调整大小至 256x256，并且经过随机裁剪至 224x224 大小。TCCA-Train 和 TCCA-Test 数据集调整图像大小至 512x512，并随机裁剪至 448x448 大小。训练过程中还使用了 50%概率的随机翻转以实现数据增强。模型训练的设置如下：batch size 为 64（TCCA-Train 和 TCCA-Test 为 32），采用动量为 0.9 的随机梯度下降（SGD）优化器，学习率初始化为 0.01 且采用余弦衰减更新学习率，均使用交叉熵损失，一共训练 30 轮。

对于色差检测任务，颜色直方图的 bin 个数设置为 20。多层感知机网络训练的参数如

下：输入层、隐藏层和输出层的神经元数量分别为 382, 25 和 1，batch size 为 128，使用标准参数的 Adam 优化器，学习率初始化为 0.006，使用二分类交叉熵损失，一共训练 2500 轮。

4.3 消融实验

Backbone	HL			TEM	PTFEM	MD-PTFEM	Concat	CBP	Accuracy	
	1	2	4						mean	std
ResNet50			✓						72.00	1.15
			✓						71.42	0.85
			✓				✓		71.52	1.67
		✓	✓					✓	73.63	0.93
		✓	✓				✓		73.19	1.05
		✓	✓				✓		73.31	1.00
		✓	✓					✓	73.75	1.32
				✓	✓		✓		73.34	1.22
				✓		✓	✓		73.45	1.12
				✓	✓			✓	74.16	1.19
				✓		✓		✓	74.33	1.13
			✓	✓		✓		✓	74.76	1.20

以下消融实验均在 DTD 数据集下进行。

纹理识别分支的消融实验 我们通过实验验证了纹理识别分支各个模块的有效性。实验结果如表 1 所示。我们选择 ResNet50 作为基线，它达到了 72.00% 的平均精度。首先将直方图层应用到主干网络第四层（HL_4），仅获得了 71.42% 的精度，接着将直方图层的输出与主干网络第四层的特征串联（HL_4+4），提升了 0.1%。该结果与[8]中分析一致，DTD 数据集中的大多数图像都具有同质纹理以及直方图层保留的局部信息，因此直方图层几乎不提供额外有用的信息。但我们将实验结果不理想的原因归结于最终的串联操作，它并没有发挥出纹理细节的统计信息与空间结构信息互补的优势。因此，最终我们将两个信息的结合方式从串联转变为 CBP（HL_4+4+CBP），正确率从 71.52% 提升至 73.90%，该实验印证了这个想法，同时也证明了 CBP 模块的有效性。接着上述实验，我们进一步将主干网络第一层和第二层的输出应用直方图层进行统计，结果与 HL_4 融合，并和主干网络第四层的输出分别执行串联（HL_1+HL_2+HL_4+4）和 CBP 操作（HL_1+HL_2+HL_4+4+CBP），准确率分别为 73.31% 和 73.75%。仅使用 HL_1 和 HL_2，且通过串联的方式与第四层输出融合，也获得 73.19% 的精度。不仅证明了直方图层的有效性，也从侧面反映了网络浅层的特征中包含丰富的纹理信息。

相比于使用直方图层统计第一层和第二层的特征，通过在模型中接入 TEM 和 PTFEM 模块，可以将性能提升至 73.34%。将 PTFEM 替换成我们改进后的 MD-PTFEM 可以进一步将正确率提升至 73.45%。实验证明了 MD-PTFEM 相较于 PTFEM 能更有效的提取纹理特征，且 TEM 与 MD-PTFEM 的组合比直方图层能更好的利用浅层的特征。最终，我们将 TEM 与 MD-PTFEM 统计于第一层和第二层特征，直方图层统计第四层的特征，将所有的统计特征串联后和第四层的特征执行 CBP 操作，该模型获得了最出色的性能，精度达到 74.76%

Method	Accuracy		Method	Reduce channels		Accuracy	
	mean	std		1×1 Conv	3×3 Conv	mean	std
ResNet50(1)	70.86	0.84	ResNet50(2+3+4)	-	-	todo	todo
ResNet50(2)	72.31	0.78		✓		74.64	0.59
ResNet50(3)	74.06	0.70			✓	74.77	0.53
ResNet50(4)	73.49	0.78		✓	✓	74.93	0.70
ResNet50+GAL(4-3)	74.6	0.58					
ResNet50+GAL(3-2)	74.52	0.61					
ResNet50+GAL(3-2+4)	74.93	0.70					

颜色分类分支的消融实验 我们证明了全局注意力层的有效性。实验结果如表 2 所示。我们使用 ResNet50 作为主干网络。将主干网络一至四层的输出直接作为颜色分类特征，分别获得了 70.86%、72.31%、74.06%和 73.49%的准确率。其中，第三层特征获得了最出色的性能，因此，该实验在一定程度上证明了，对于带纹理的图像，并非越深层的特征富含越丰富且准确的颜色信息。因此，我们利用第三层的特征指导第二层的特征，获得了 74.52%的准确性，尽管利用第四层的特征指导第三层的特征可以获得更好的性能，但利用第二层的特征仅获得了 72.31%的准确率，这有力的证明了全局注意力层的有效性。最终，我们利用第三层的特征指导第二层的特征，并与第四层的特征融合，获得了最好的性能。

我们进一步研究了全局注意力层中对低层特征降维的操作，进行了如下四个实验：（1）不降维，（2）使用 1x1 的卷积进行降维，（3）使用 3x3 的卷积进行降维，（4）使用 1x1 的卷积降维，**再使用 3x3 的卷积调整特征**。结果如表 3 所示，**不降维仅获得了 xxx 的精度**，而加入任意一种降维操作均能使模型的性能得到有效的提升，其中 1x1 与 3x3 卷积结合使用的降维方法性能最优，准确率可达 74.93%。

4.4 与最前沿的方法比较（**TODO 差 FMD 的实验**）

Method	Source	BackBone					
		ResNet18			ResNet50		
		DTD	MINC-2500	FMD	DTD	MINC-2500	FMD
DeepTEN	CVPR17	-	-	-	69.6	81.3	80.2±0.9
DEPNet	CVPR18	-	-	-	73.2	82.0	-
LSCNet	PR19	-	-	76.3	-	-	81.2
MAPNet	ICCV19	69.5±0.8	-	80.8±1.0	76.1±0.6	-	85.2±0.7
DSRNet	CVPR20	71.2±0.7		81.3±0.8	77.6±0.6	-	86.0±0.8
HistNet	ArXiv20	-	-	-	72.0±1.2	82.4±0.3	-
CLASSNet	CVPR21	71.5±0.4	80.5±0.6	82.5±0.7	74.0±0.5	84.0±0.6	86.2±0.9
FENet	NIPS21	69.59±0.04	80.57±0.1	82.26±0.29	74.20±0.10	83.98±0.14	86.74±0.19
Ours		71.32±0.89	80.66±0.47		74.76±1.20	83.36±0.40	

我们的模型与 8 种基于 CNN 的纹理识别方法进行了比较，其中包括 DeepTEN、DEPNet、

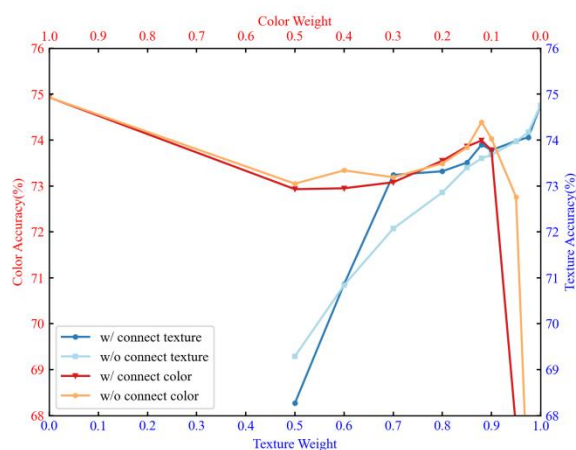
LSCNet、MAPNet、DSRNet、HistNet、CLASSNet 和 FENet。表 4 列出了 3 个数据集的结果。比较的结果尽可能引用现有文献（主要来自[15]），否则留空。使用红色和绿色区分最好的结果与具有竞争力的结果。

ResNet18 作为主干网络。我们的模型在 MINC-2500 数据集上取得了最好的结果，与第二好的表现相比，精度提升了 0.11%。对于 DTD 数据集，我们模型的平均精度超过了 MAPNet 和 DSRNet（这两个方法在以 ResNet50 为主干网络的实验中表现出明显的优势），且与 CLASSNet 相当，仅相差 0.18%。

ResNet50 作为主干网络。尽管使用 ResNet50 作为主干网络竞争力有所下降，但仍具竞争力。在 DTD 数据集上的结果超过了最近提出的 CLASSNet 和 FENet，仅次于 MAPNet 和 DSRNet。我们的模型在 MINC-2500 和 FMD 数据集上的结果没有显示出明显的优势，其结果也可以预见。我们的方法利用了 ResNet 网络前两层的特征，ResNet50 相对于 ResNet18 在前两层并没有大量增加卷积层数，而是在此之后通过增加通道数以及卷积层数，从而使网络的深度增加，获得更具有区分性的特征。因此，前两层特征并没有发生很大的变化，导致 PTFEM 和 MD-PTFEM 模块无法提供更多额外的纹理统计特征，这也解释了为什么 ResNet50 性能的改进不如 ResNet18。

TODO: 我们的模型更适合轻量网络...

4.5 多任务模型实验



MOO-MTL		w/ connect		w/o connect	
Texture	Color	Texture	Color	Texture	Color
1	0	-	-	74.76	-
0.975	0.025	74.06	48.57	74.17	66.03
0.95	0.05	73.98	67.73	73.97	72.76
0.9	0.1	73.78	73.78	73.68	74.03
0.88	0.12	73.90	73.99	73.60	74.39
0.85	0.15	73.51	73.86	73.40	73.84
0.8	0.2	72.32	73.55	72.86	73.49
0.7	0.3	73.24	73.08	72.07	73.19
0.6	0.4	70.86	72.95	70.84	73.34
0.5	0.5	68.27	72.93	69.29	73.05
0	1	-	-	-	74.93
MOO-MTL		73.78	73.76	73.52	73.76
SLAW		72.20	73.32	72.11	73.41

我们使用 ResNet50 作为主干网络比较了以下三种平衡任务之间训练的方法（1）LW：最小化两项任务损失的加权线性组合。（2）MOO-MTL[16]：找到多目标优化问题的一个帕累托最优解。（3）SLAW[17]：使用 Crawshaw 等人提出的 Scaled Loss Approximate Weighting。

TODO: 介绍一下我为什么要选择这两个损失进行比较，有什么优势

从图 1 的多任务框架中我们可以看到，颜色分类任务与纹理识别任务间存在一条融合两者特征的连接，我们分别实验了有这条跨任务连接与没有跨任务连接对各任务精度的影响。表 5 显示了我们多任务实验的结果。与单任务训练相比，使用以上任一方法联合训练两个任务都会导致两个任务的精度有所下降，其原因正如 3.3 节开头所描述的那样，纹理特征中的颜色信息几乎为零，因此两个任务并不相关，甚至是竞争的，这导致了两个任务的准确率受损。通过细粒度网格搜索最佳权重组合的方法（LW）获得了优于 MOO-MTL 与 SLAW 的性能，而 MOO-MTL 的性能全面领先 SLAW。图 8 记录了不同损失权重对两个任务精度的影响。对于纹理识别任务，含有跨连接的多任务模型整体性能更优。而对于颜色分类任务则相反。其中可能的原因是：颜色特征为纹理识别提供了额外的信息，使得纹理特征更具区分性，从而改善了纹理识别的精度。但两个具有竞争关系的任务，由于跨连接的存在，导致纹理识别的结果直接影响了颜色分类分支的参数，使得颜色分类精度进一步下降。从图中可以明显看出，当纹理任务与颜色任务损失的权重为 0.88 和 0.12 时，两类多任务模型均获得了最优的性能，由于纹理识别为主任务，因此我们在设计上选择了纹理识别性能更优的跨连接多任务模型，其在纹理与颜色任务的精度可达 73.90% 与 73.99%，该精度与单任务训练相比，下降的幅度均小于 1%。为获得额外的颜色信息，性能损失的程度完全在可接受范围内。

4.6 色差检测

Method	Texture Descriptor		Color Descriptor	Flag Descriptor	Accuracy	
	Texture	Color			mean	std
ResNet50	-	-	-	-		
SVM	✓	✓	✓	✓		
MLP	✓					
	✓	✓				
	✓		✓			
	✓		✓	✓		
	✓	✓	✓	✓		

1. 单纯纹理（单任务 存储）
2. 多任务（纹理 颜色 分开存储）
3. 颜色描述符（存储）
4. Flag 描述符（存储）

4.7 实时性*（到时候效果不好就删除）

5 结论

参考文献

- 1 关于瓷砖产生色差的原因分析
- 2 英国的 Stuface Inspedion 公司
- 3 Automatic Grading of Ceramic Tiles Using Machine Vision
- 4 Automatic Color Grading of Ceramic Tiles Using Machine Vision
- 5 Colour Grading of Randomly Textured Ceramic Tiles Using Colour Histograms
- 6 Perceptual Correction for Colour Grading of Random Textures
- 7 Colour tonality inspection using eigenspace features
- 8 基于 CCD 的瓷砖色差与尺寸检测研究 2007
- 9 瓷砖色差在线检测系统的研究 2009
- 10 Real-Time Surfae Grading of Cerami Tiles
- 11 墙地砖颜色的自动分类研究
- 12 瓷砖色差在线检测系统的研究
- 13 墙地砖缺陷的智能检测技术研究
- 14 基于共生矩阵和颜色特征的墙地砖缺陷自动检测的研究
- 15 TEX-Nets: Binary Patterns Encoded Convolutional Neural Networks for Texture Recognition
- 16 Fully convolutional networks for semantic segmentation
- 6 Compact Bilinear Pooling
- 7 Learning Statistical Texture for Semantic Segmentation
- 8 Histogram Layers for Texture Analysis
- 9 Random feature maps for dot product kernels
- 10 Fast and scalable polynomial kernels via explicit feature maps
- 11 Describing textures in the wild
- 12 Material recognition in the wild with the materials in context database
- 13 What can you see in a brief glance?
- 14 Deep residual learning for image recognition

- 15 Deep Texture Recognition via Exploiting Cross-Layer Statistical Self-Similarity
- 16 Multi-Task Learning as Multi-Objective Optimization
- 17 SLAW: Scaled Loss Approximate Weighting for Efficient Multi-Task Learning