1. What is Channel?
2. What is feature map?
3. What is pool?
4. End-to-end training?

**Abstract:** 设计一个以相同的方式作用在大的窗口上的匹配代价函数是困难的，代价函数不够智能来舍弃和目标像素无关的信息，导致了不想要的人造误差。这篇论文，提出了一个新奇的CNN模型来用一个大的窗口来学习立体匹配代价。与具有跨度的的传统池化层不同，提出的逐像素的金字塔池化层能够覆盖大的区域而不损失分辨率和细节。因此，学习的匹配代价函数能够成功地利用从大的区域学到的信息而不会引入增肥效果（fattening effect？）。尽管存在弱纹理，深度不连续性，照明和曝光差异，但所提出的方法是稳健的。它在Middlebury benchmark上实现了接近最好的表现。

**Introduction：**大多数立体匹配方法，在使用一定的先验知识来全局或者局部优化整个匹配代价卷之前，首先计算每个像素的以一个确定视差的匹配代价。数十年来，许多研究者聚焦在第二个步骤，即设计一个好的先验函数和优化它。很少有研究关注设计和选择一个更好的匹配代价函数。

一种最广泛使用的匹配代价函数是像素匹配代价函数。结合复杂的（sophisticated）先验模型，它有时会产生好的结果，尤其能很好的保存视差不连续区域附近的细节结构。然而，这种函数在图像包含弱纹理区域或重复纹理区域是失败的。在这种情况下，基于窗口的匹配代价函数，例如CENSUS和SAD，产生了更可靠和有区别性的测量。基于窗口的匹配代价函数的严重缺点是它在视差不连续区域的不稳定性。

处理这种权衡的一种方式是针对输入的模式使基于窗口具有多功能性（versatile）。关键思想是使匹配模板的形状自适应以使它能够丢弃和目标像素无关的像素的信息。然而，在精确的匹配前知道背景像素是困难的，使它成为一个“鸡和蛋”问题。因此，使用CNN是合适的，因为它可以自动地为每个输入模式学习合适的（proper）模板形状。

然而，现存的模型，是基于类似于AlexNet或者VGG网络传统的CNN架构，它们都是针对图像分类问题进行了优化而不是针对图片匹配。这些结构包含不同的卷积层，每个后面跟着修正线性单元（ReLU）和带跨度的池化层。使用这些结构用来匹配的一个限制在于增大用来比较的块（patches）的尺寸的困难。块的有效尺寸直接关联CNN的感受域的空间范围，能够被增加通过（1）包括一些跨步（strided pooling/convolution layers）的池化/卷积层；(2)在每一层使用更大的卷积核；或者（3）增加层的数量。但是，使用跨步池化/卷积层会使结果下采样，从而丢失细节。尽管分辨率能够通过应用分数跨步（fractional-strided）卷积恢复，重建小的或者薄的（thin）结构仍然困难如果一旦他们在下采样后丢失。增加内核的大小也是有问题的，因为表示较大模式所需的特征图的数量显着增加。而且，一项之前的研究报道，重复使用小的卷积并不总是导致大的感受域。

这篇文章通过提出一种新颖的CNN模块来学习一个更好的匹配代价函数，为文献做出了贡献。这种模块是一种新颖的池化模块能够使CNN看到一个更大的区域而不丢失细节，并且能够在测试时间内不增加计算的复杂性。实验显示，使用提出的模块提高了基线网络（baseline network：MC-CNN？）的表现，在Middlebury benchmark上展现了有竞争力的结果。

**Related Works：**

鉴于引入了具有真值视差图的高分辨率立体数据集，已经进行了许多尝试使用机器学习算法来学习匹配代价函数。最引人注目的结果是通过使用CNN获得的（MC-CNN和LCP-CNN）。MC-CNN提出的结构采用了一个小的11×11的窗口，并且没有使用池化处理它。因为窗口的大小是有限制的，计算的匹配代价卷是有噪声的。因此，通过使用交叉代价聚合（cross-based cost aggregation），半全局匹配（SGM）和另外的精化程序被后处理。另一方面，LCP-CNN中的方法使用了多个池化层和空间金字塔池化（SPP）来处理更大的块。然而，结果展示了由于池化引入的信息损失导致的一个胖效应（fatting effect？）。

这篇论文主要的贡献是提出了一个新奇的池化模式，可以从一个大的感受野来处理信息而不会丢失细节。最近，数个不同的尝试被提出来完成相同的目标在语义分割的上下文中（context）。这些模块将来自高级层和较低层的特征图结合在一起，目的是正确地沿着像素级的细节对其对象级（object-level）的信息进行校准。尽管这个方法可以成功地校准大对象的边界，它固有的限制是它不能在最后的输出中恢复小的对象一旦它们在由于多次使用池化的抽象过程中丢失。在同一个环境下（context），FlowNet结构能够通过使用低层级（lower-level）的特征图将粗糙级流（coarse-level flow？）上采样到原始比例。然而，由于高层级的特征图的低分辨率，它不能恢复被隐藏的极端的流元素（flow elements?）。

和当前工作最相关的结构在【24】（SPP）中被提出。不像其他方法，SPP网络排除了卷积层之间的池化层。代替的是，它首先通过多次级联（cascading）卷积层计算了高度非线性特征图，然后通过在不同尺度（scale）池化它们产生了高层级（high-level）和中层级的信息。通过将原始特征图和在不同尺度下下池化的特征图放在一起，SPP网络能够结合来自不同层级（level）的特征而不丢失细节。尽管之前【14：LCP-CNN】中提到的立体匹配方法使用了SPP，由于它在卷积层之间采用了传统的池化层，所以丢失了细节信息。

Architecture of the neural network

提出的结构采用两个输入块并产生对应的匹配代价。在接下来的子部分，新提出的模块首次被介绍。然后整个网络的细节结构将会被展示。

1. 每个像素的金字塔池化（per-pixel pyramid pooling）

CNN中池化层的使用被认为是可取的，因为它在图像分类任务中的有效性。虽然据报道最大池化层的使用在空间转换中提供了额外的不变性，但最重要的增益来自于特征图的下采样。通过执行大于1的跨度的池化，池化后输出的特征图被缩小。CNN输出的最后的尺度（scale）关于池化层的数量成指数下降。考虑到没有关于池化层的参数存在，这个方法是一个有效的方式来扩宽CNN的感受野，且不会增加参数的数量。跨步池化层的缺点是网络损失了原始特征图的细节因为池化的应用。因此，看到更大的范围和保留小的细节存在着一种权衡（trade-off）。

受【24：SPP-CNN】思想启发，我们提出了一种新的池化模式来克服这种权衡（trade-off）。代替使用一种小的带跨度的池化窗口，一个大的池化窗口被用来实现感受野（receptive field）的想要的尺寸。一个大的池化窗口的使用能够导致细腻（finer）的细节的损失。因此，多个不同窗口尺寸的池化被执行，输出被连接（concatenated）来产生新的特征图。得到的特征图包含来自粗燥-精细（coarse-to-fine）尺度的信息。多尺度池化操作不带跨步（without strides）地为每个像素所执行。

我们叫整个过程为“每个像素的金字塔池化”（per-pixel pyramid pooling:4P）,形式上被定义如下：

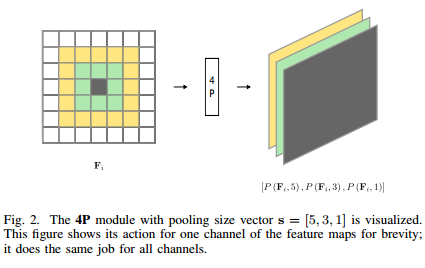
, (1)

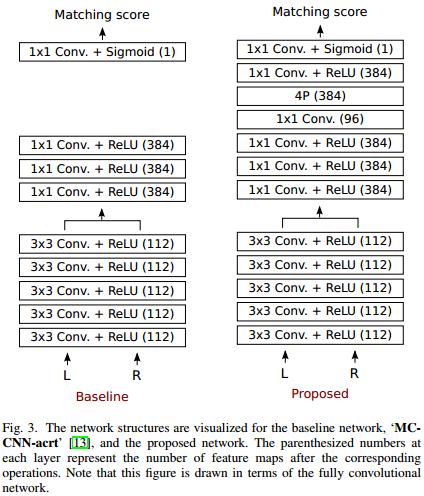
其中，s是有M个元素的向量，是大小为，步长为1的池化操作 。这个模型的结构如图2所示。

1. 提出的模型

为了验证提出的模型的有效性，我们训练和测试了有和没有4P模块的CNNs。基线架构（baseline architecture）被选为MC-CNN-acrt。提出的架构中的4P模块通过使用尺度向量s=[27, 9, 3, 1]来构建。两个CNNs结构在图3中可视化。

Implementation details





为了公平的比较，我们跟着MC-CNN-acrt中的细节训练提出的结构包括一些下面提到的例外。首先（first），训练块的大小变成了37×37。而且（furthermore），我们仅微调了图3中提出的架构的最后三个1×1卷积层的参数。较早层的参数是借自预训练的MC-CNN-arct网络。在我们的实验中，这导致了一个更好的表现比起随机初始化的端到端（end-to-end）的训练网络。再者（moreover），利用预训练的特征来训练一些卷积层更容易。我们已经运行了总共4轮的训练，最后两轮用一个从0.003到0.0003的下降的学习率来运行。

在测试阶段，我们也用了和MC-CNN-acrt同样的后处理管道（pipeline）。后处理管线包括使用CBCA和SMG，并且视差图被优化来有连续的值，并且进行了中值滤波和双边滤波。

**Experiments**

为了验证提出的4P模型的有效性，我们比较了基线和提出的网络的结果。表现被测量通过使用Middlebury benchmark上的“training dense”集。定量的结果被简短地总结在表一中，使用平均误差。所有的实验被执行通过使用Inter core i7 4790K CPU和一个独立的Nvidia Geforce Titan X GPU。

提出的方法优于基线架构不管有没有使用后处理。使用4P模块的好处是明显的当视差图被获取通过使用像素方式的赢者通吃（pixel-wise winner-takes-it-all）规则而没有任何的后处理。考虑到数据集中的图像包含许多弱纹理区域，小尺寸11×11的窗口不能在没有后处理的帮助下从错误的匹配中分辨正确的匹配。在另一方面，提出的结构有效地看到更大的窗口，37×37，通过在最后的决定层前插入4P模块。

不那么直截了当去理解为什么提出的架构在后处理之后依旧表现得比基线好。从这个意义上将，值得提到提出的方法的后处理的最好的参数设置和基线（MC-CNN-acrt）有很大的不同。和原始参数设置最显著的改变是，我们使用了更少数量的CBCA，它意味着提出的架构中的CBCA的多重（multiple）使用变得冗余。从这个事实出发，我们可以解释4P模块的作用为自适应（adaptive）局部特征聚合。比较手动设计的算法如CBCA，临近像素对特定像素的影像是自动学习的并且它可以与代价方程本身共同训练。而且，像素之间信息的交换在特征空间（feature space）中进行，特征空间包含着更丰富的上下文信息（contextual information）比起最后的代价卷空间。

注意在基线上的提高很明显不是由于额外层的使用，也不是由于更多参数的使用，因为基线作者已经展示了额外使用全连接层意义不大。使用两层额外的全连接层导致了大约1.90%的提高，然而使用4P模块导致了平均误差21.42%的提高。

提出方法的主要贡献在于通过检查（inspecting）一个更大的区域来学习一个更少模糊的匹配代价函数。图4展示了提出的网络确实做得更好在弱纹理区域比起MC-CNN-acrt。每个数据集定量的和定性的结果，包括在“test dense”集中的，在Middelbury benchmark网站上都能得到。

**Conclusions**

观察一个大区域来估计稠密像素的对应关系是有必要的以完全利用纹理信息来实现更少模糊和更精确的匹配。一个传统的匹配代价函数是失败的因为与目标像素在同一表面上的

相邻像素是未知的。在这篇论文中，一个新的CNN模块被提出来来使CNN结构处理一个更大的图像块而不丢失小的细节，这使得它能为大尺寸的窗口学习一个智能的匹配代价函数。学到的代价函数能够为弱纹理或者重复纹理区区分出错误的匹配，并且能够很好地保存视差不连续性。学到的代价函数实现了又竞争力的表现在Middlebury benchmark。

