指导教师：杨涛　　提交时间：2016/3/13

CVPR2015 Paper Translation

No: 1

姓名：郑添喜

学号：2013302589

班号：10011305

一种动态卷积层的短距离天气预报

Benjamin Klein, Lior Wolf and Yehuda Afek
The Blavatnik School of Computer Science
Tel Aviv University
beni.klein@gmail.com, wolf@cs.tau.ac.il, afek@post.tau.ac.il

摘要

我们提出了一种名为“动态卷积层”的深层网络层，它是对卷积层的一个推广。传统的卷积层使用在训练时被学习并在测试时保持不变的滤波器。相比之下，动态卷积层使用在测试时从输入到输出都在变化的滤波器。这是通过学习一种用于对滤波器的输入进行映射的函数实现的。我们在短距离天气预报上使用动态卷积层，并且展示它与其它基线比较的提高。

1.介绍

深入学习和具体的卷积神经网络(CNNs)[16]在解决各种计算机视觉的应用始越来越受欢迎。让 CNNs 区别于其他的神经网络它是利用卷积层。这一层通过使用一组滤波器卷积上一层的特征映射来计算输出特征映射。这些滤波器是卷积层的唯一决定因素并且通常使用反向传播算法在训练时被学习。

近年来，CNNs 已经在各种挑战性问题上实现了最先进的成果，如：目标识别、目标定位、肿瘤检测、人脸识别和场景标签。
[15,5,24,25,26,11]。鉴于 CNNs 的成功，我们决定它们在短距离天气预测上的表现。在这项任务中，一个接收一系列的雨雷达图像（图 1），每 10 分钟采取一个图像，目标是预测序列中的下一个图像。虽然我们理解在识别问题中卷积层的关键作用，但是我们认为，不同的网络架构可以更适合短距离天气预报任务。可以观察到，按序列排布的雷达图片通常可近似为前一个图片的转换，根据所有序列的运动行为，它表明了新的转换序列中最新图片的卷积层的需求。

出于这一观察，我们提出了一个称为“动态的卷积层”的新的深层网络层，它推广了传统的卷积层。类似于卷积层，动态卷积层从前一层提取特征映射，然后用滤波器卷积它们。新颖性在于，动态卷积的滤波器并不是这一层关键参数，而是它们获得的映射输入一系列滤波器的任意深度的一个子网的输出（图 3）。

在这项工作中，我们提出了动态的卷积层，并用于的短期天气预报。我们把结果和其它基线比较，包括不使用动态卷积层的卷积神经网络。通过使用新的卷积层，与其它基线比较，包括传统 CNN，我们获得了表现提升的结果。

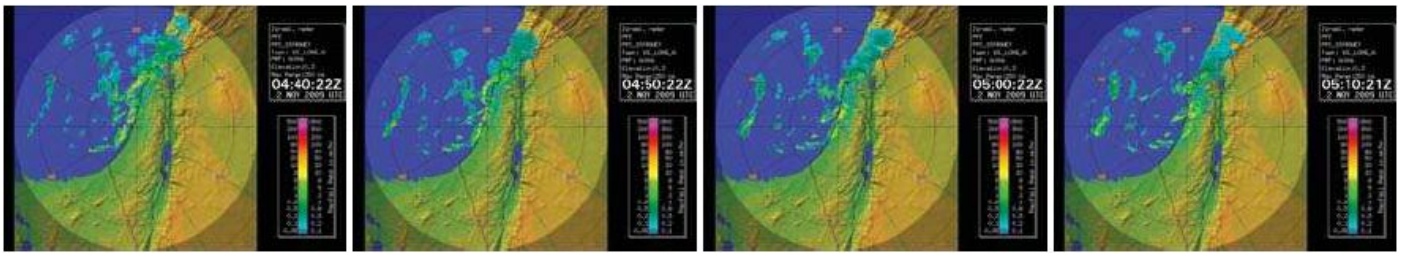

图 1.来自 Tel Aviv Dataset 的 4 张连续的未预处理过的雷达图片。相邻两张图片间隔 10 分钟。

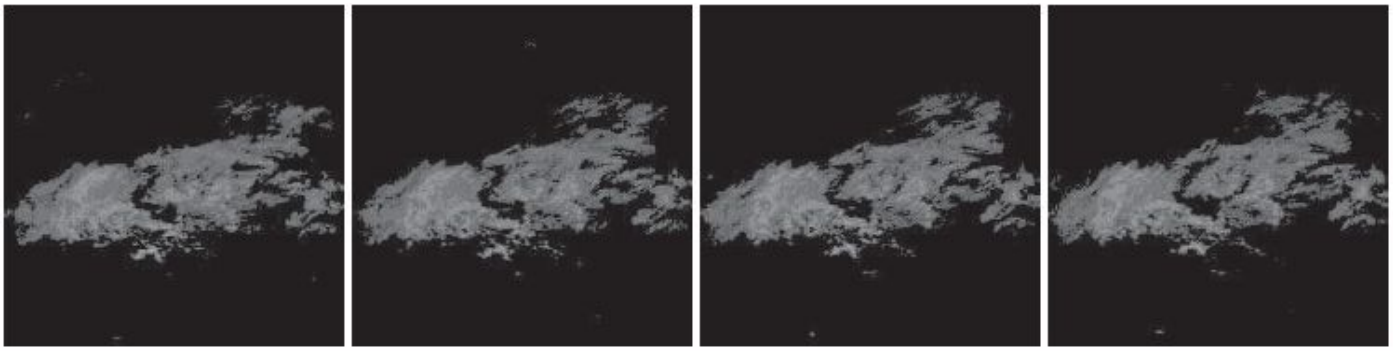

图 1.来自 Davenport Dataset 的 4 张连续的已预处理过的雷达图片。预处理时，一个恒定的颜色变换函数被应用于每个图像，以改变由颜色强度编码的云层强度。

2.相关工作

在最近几年中，一些论文建议修改卷积层。在[17]，Lei 等，提出一种新的名叫“网络上的网络”（NIN）的深度网络架构。这种结构在传统的卷积层与多层感知器上代替线性滤波器。使用这种新的结构，他们在 CIFAR-10 和 CIFAR-100 取得了最先进的表现。这工作不同于我们的工作主要在两个方面：第一，它取代了卷积与多层感知器；第二，新的块被依次输入来替换卷积层，即，新的块作为卷积层替换具有相同的输入层和相同的输出层。在我们的例子中，我们坚持以卷积算子，它适用于我们在意的图像合成的任务（NIN 用于分类）。我们使用分离的子网计算卷积层，子网能采用任意之前一层的输出作为输入。

相比于我们的工作，Cohen 等[6]提出了另一个 CNNs 的推广。这种推广增加了掩蔽权重到相似性层，从而推广了传统的卷积层。与我们的工作相比，这个权重是在优化过程中学习的，而不是在评估新样本是计算出的。此外，每当相似层计算线性相关性和非函数距离时，这个权重都能很简单得被吸收到滤波器。在我们的例子中，动态性的滤波器无法通过传统的网络表达。

每一层的动态卷积层，来自上一层的输入和滤波器从另一层得到的输入的相互作用具有乘法性质。在文献中，乘性相互作用出现在计算两张运用自动编码的图像[1,9]的相似性的时候，或者在为了学习采用 Boltzmann 机进行动作识别[27]的两个连续帧的相互作用而计算相似性的时候。乘法的相互作用也用于连接带有网络输入[20,19]的隐藏的变量，其中隐藏的变量被淘汰使用边缘化。在我们的网络中，这两种路径导致的乘法相互作用来自使用前馈方式得到的网络输入。这输入并不分为单独传播的 2 个图像或两个帧。

乘性的相互作用也出现作为的 Sigma-Pi 单位[21]的一部分，其中前一层的输入的被乘以创建非线性相互作用。相比我们的工作，很早以前的文献考虑相同的输入层的不同元素的直接作

用，并在简单的人工神经网络背景下探讨。在[18]中可发现更多关于乘性相互作用的先前工作的参考文献。

天气预测 作为一种应用，我们采用气象监测雷达图像，以短期预测未来雨雪的位置和强度。这样的预测支持非正式的用法，如帮助用户决定是否要骑自行车去学校，和社会层面的紧急警报，如预测山洪爆发。之外，输出预测可用于交通预测和被集成到机场天气系统。

关于这一主题的文献似乎更侧重于天气雷达的物理性质[28]，在模型校准的输出与土地测量[3]，并在整合所获得的数据，使用商业产品，与其他气候模式[7]。虽然尚不清楚商业产品如何工作的，但是很明显，在计算机视觉的最新进展的任务中，雨天气预测的使用仍然有很大程度上缺乏。

3. 动态卷积层

在这一节中，我们将描述动态的卷积层，这是一个扩展的卷积层。常规的卷积层解释在 3.1，并在 3.2 解释动态卷积层。讨论探讨了层在输入，输出，向前和向后通过的不同。

3.1 传统卷积层

卷积层接收一个单一的输入-来自上一层的特征映射。卷积层利用卷积滤波器和输入来计算特征映射作为输出。这些滤波器是卷积层的关键，并且在训练时使用反向传播[22]被学习。在测试时，它们将会被固定且不会改变。

前向传递 通常是在样本 T 中学习。我们把 x_i^t 表示样本 t 的第 i 个输入的特征映射，y_j^t 表示样本 t 的第 i 个输出的特征映射。滤波器表示为 k_{ij}。在卷积层的前向传递中，输出卷积映射是用卷积算子（表示*）计算的：

$$y_j^t = \sum_i k_{ij} * x_i^t \tag{1}$$

后向传递 在后向传递中，卷积层利用 x_i^t 计算网络的损失函数 l 的梯度:

$$\frac{\partial l}{\partial x_i^t} = \sum_j \left(\frac{\partial l}{\partial y_j^t} \right) \overset{*}{-} \left(k_{ij} \right) \tag{2}$$

$\overset{*}{-}$ 是用零填充的卷积。作为后向传播算法的一部分，梯度 $\frac{\partial l}{\partial x_i^t}$ 的价值被传递给计算 x_i^t 的上一层。损失函数的梯度关于 k_{ij} 的计算：

$$\frac{\partial l}{\partial k_{ij}} = \frac{1}{T} \sum_t \left(\frac{\partial l}{\partial y_j^t} \right) * \left(\tilde{x}_i^t \right) \tag{3}$$

$\tilde{x}_i^t$ 是 x_i^t 的行/列翻转。计算完 $\frac{\partial l}{\partial k_{ij}}$ 之后，这一层的系数 k_{ij} 通过使用以下公式更新：

$$k_{ij} = k_{ij} - \alpha \cdot \frac{\partial l}{\partial k_{ij}} \quad , \tag{4}$$

α 是学习比率。

3.2. 动态卷积层

不同于传统卷积层，动态卷积层有两个输入。第一个输入是来自上一层的特征映射，第二个是滤波器。特诊映射得自于输入，通过跟随一个子网 A。滤波器是使用一种分离的输入的卷积性网 B 的结果。卷积层的输出是通过卷积滤波器过滤来自上一层的的特征映射计算出来的，相同的方法应用在卷积层，但是这儿的滤波器是输入函数，因此在测试时一个样本和另一个不同。整个

系统是一个定向非循环的卷积层图，所以使用后向传播算法训练。

前向传播 在前向传播中，网络 A 计算特诊映射，该特诊映射将会作为第一个输入给动态卷积层，并且被分离的子卷积网络 B 计算滤波器，该滤波器将会作为第二个输入给动态卷积网络(图 3)。让 x_i^t 作为第 i 个样本 t 的特诊映射输入，让 k_{ij}^t 作为第 t 个样本的 ij 输入核，让 y_j^t 作为样本 t 的第 j 个输出特征映射，然后再动态圈基层的前向传播中，输出特征映射的计算公式如下：

$$y_j^t = \sum_{ii} k_{ij}^t * x_i^t \tag{5}$$

注意：不同于传统卷积层，动态卷积层的每一个样本都有一个不同 k_{ij}^t 核。

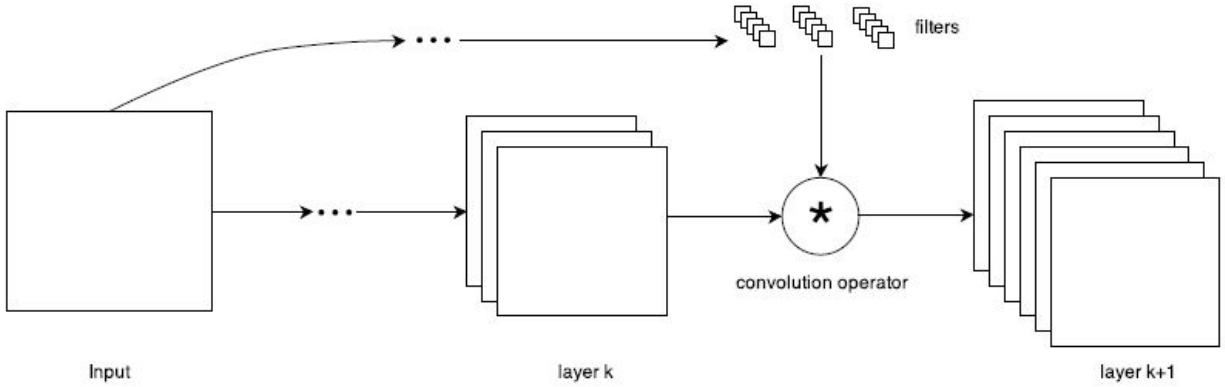

图 3.动态卷积层。通过卷积滤波器计算第 k 层特征映射得到第 k+1 层。这些滤波器是在输入中应用卷积网络的结果。

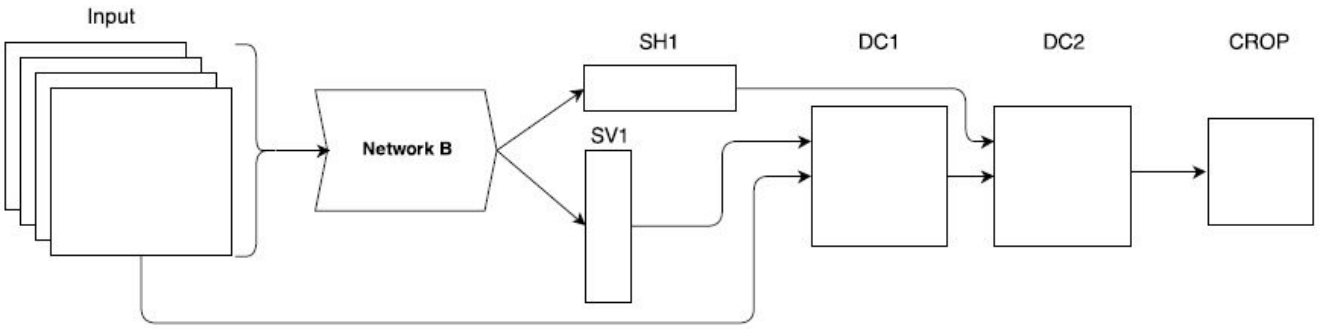

图 4.网络结构。网络 B 是一个使用动态卷积层计算滤波器(H1 和 V1)的子网。SH1 是使用 softmax 函数在 H1 的结果，SV1 是使用 softmax 函数在 V1 的结果。DC1 是动态卷积层，该层提取序列中的最新的图片，并利用 SV1 卷积它。DC2 是动态卷积层，该层提取 DC1，并利用 SH1 卷积它。

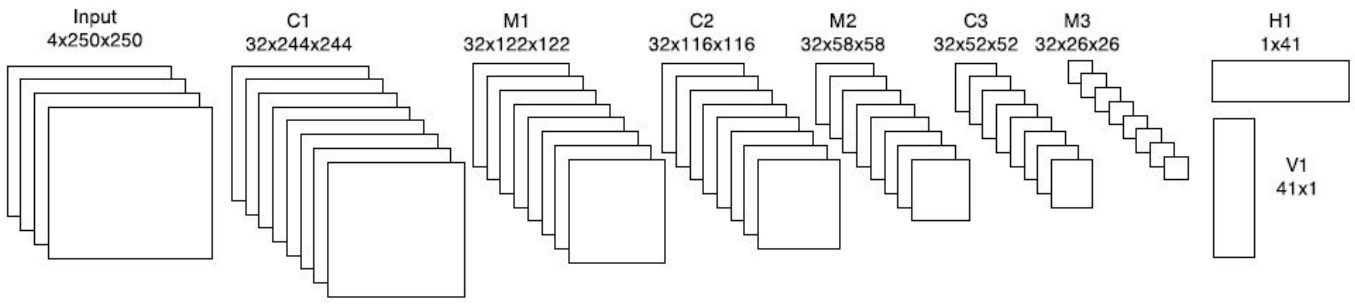

图 5. 整个图片整体网络 B 的结构。C1,C2 和 C3 是传统卷积层，M1,M2 和 M3 是最大池的层。每一个最大池层后面，我们使用 tanh 非线性激活函数。H1 和 V1 都通过一个完全连接的层连接到 M3。

后向传播 在后向传播中，动态卷积层利用 x_i^t 计算损失函数 l 的梯度，公式类似：

$$\frac{\partial l}{\partial x_i^t} = \sum_j \left(\frac{\partial l}{\partial y_j^t} \right) \overline{*} \left(k_{ij}^t \right) \tag{6}$$

梯度 $\frac{\partial l}{\partial x_i^t}$ 的价值被传递给产生 x_i^t 的的网络 A。

并且，类似于传统卷积层，关于 k_{ij}^t 的损失函数的梯度的计算公式如下：

$$\frac{\partial l}{\partial k_{ij}^t} = \left(\frac{\partial l}{\partial y_j^t} \right) * \left(\tilde{x}_i^t \right) \tag{7}$$

相比于卷积层，k_{ij}^t 不是卷积层的参数，它们是输入 t 的一个函数，t 是来自于网络 B 的上一层。因此梯度

$\frac{\partial l}{\partial k_{ij}^t}$ 的价值被传递给作为后向传播算法中计算 k_{ij}^t 的卷积层。

4. 全图合成

全图合成方法得到 4 个 250×250 雷达图片以作输入，并输出序列中预计的下一个图片。预估是使用包含动态卷积层的 DNN。在实践中，预测包含序列中下一个图片的 200×200 个中心斑。因为边界时显著被云层影响的，云层在先前的雷达图片中没有被看见。

网络架构 全部的架构展示在图片 4 中。这 4 张 250×250 的雷达图片作为 4 个频道输入进带着 32 个 7×7×4 的滤波器的传统卷积层(C1)。由此产生的 32 个特征映射随后被传递给最大池层(M1)，该层有最大超过 2×2 的步幅为 2 的空间块。然后输入到带着 32 个 7×7×32 的滤波器的传统卷积层(C2)。产生的 32 个特征映射被传递给最大池层(M2)，该层有最大超过 2×2 的步幅为 2 的空间块。接着输入到带着 32 个 7×7×32 的滤波器的传统卷积层(C3)。产生的 32 个特征映射被传递给最大池层(M3)，该层有最大超过 2×2 的步幅为 2 的空间块。

直觉 网络架构赋予 SV1 和 SH1 意义。这两个向量中的每一个都作为可能性的向量-它们是整数且和为 1(应用 softmax 函数的结果)。因为 SV1 被用作卷积序列中最后一个图片垂直滤波器，它在垂直轴中进行转换。同样，SH1 是一个水平卷积层，用于卷积结果特征映射，从而在水平轴上进行转换。因此，网络学习映射 4 个序列帧的输入并产生可能的卷积向量，用于预测横轴和竖轴上的适当转换，这将会使序列中最后一个图片转换成看不见的下一个图片。

5. 块合成的块

全图合成方法在案列中表现很好，案列中序列的所有对象都被相同数量转换。为了处理更复杂的场景运动，一个块合成的块被提出。用这种方法一个 DNN 在雷达序列块中(小图区域)被训练。网络的输入是 4 个 70×70 的雷达图片输入，并且它的输出是下一个序列图片块中的 10×10 中心块。为了计算序列中的下一个图片，DNN 被用于滑动窗口方式。滑动窗口方式过去已经被成功运用，例如在检测乳腺组织学有丝分裂图像[5]中的应用。

网络架构 该网络架构非常相似于全图合成，因此我们仅描述两者的不同。由于相同的输入尺寸，我们去除最后一个卷积层(C3)和最后一个最大池层(M3)。还有，产量层(CROP1)采用(DC2)的 10×10 中心块。

6. 结果

动态卷积层是先进的，测试过的，并且比较于可替换的三个非常大的小范围天气预测数据集。我们在 6.1 节描述数据集并且预处理步骤在 6.2 节。训练和处理的细节在 6.3 节。我们比较的方法在 6.4 节。最后，在 6.5 节完成三个数据集的所有方法的精度的比较。

6.1 数据集

我们使用三个数据集。第一个数据集包括采自以色列 Tel Aviv 雷达图片，第二个数据集包括采自爱荷华州 Davenport 的雷达图片，第三个数据集包括采自密苏里州 Kansas 城市的雷达图片。每一个数据集被分开训练、验证和测试以便每一集都包含采自不同年份的序列-为了防止污染。许多天中，许多序列是空的或者近乎空的。这种序列很容易预测，因此从基准过滤掉。

所有的数据集将会被下采样以创造易处理的实验。3 个训练集包括 32000 个序列，每一个验证集包括 4200 个雷达数据集，每一个测试集奥括 3200 个序列。

6.2 预处理

Tel Aviv 雷达图片集包括作为背景图片的地图(图片 1)。为了建立有意义的模型，除去背景被应用于该集的每一个图片。

接下来，对于所有的三个数据集，我们应用恒定颜色转换函数在每一个图片上，为了改造描绘云层强度的颜色地图成图片强度(图片 2)。注意：简单的运用灰度尺标转变图片并不会产生理想的结果，因为灰度尺标转变雷达图片的彩条不是单调的。我们收获和测量图片成 250×250 的像素。

6.3 处理细节

使用开源的 Caffe 图书馆[14]处理动态卷积层。DNN 中的所有权重被初始化，根据 Xavier 设定初值[13]。训练使用随机梯度下降法(SGD)，用 0.001 的学习比率和 0.9 的动量。

6.4 基线方法

动态卷积层被比较于几个可替换的技术，它们在后文介绍。注意：线形回归和 CNN 方法不会用于整个图像，仅用于块。这是因为全图的更大的输出尺寸方法，该方法在模型中有着显著增加的大量参数，使这些模型不太可行。

最后一帧 预测和输入序列有很强的相互关系，因此能简单的运用最后一帧作为预测。因为最后一帧被期望最接近预测帧，这基线变现得相当好。

全局运动估计 一个全局运动向量（dx, dy）使用黑盒子运动估计推算出来，它是 opencv 的 Keypoint Based Motion Estimation 函数[4]。最后一个图片根据这个向量转变，结果就是预测。注意：有时当因为各种原因不能成功估计

Method	Tel Aviv Dataset	Davenport Dataset	Kansas City Dataset
Last Frame	20.059±0.536	258.818±2.552	241.392±2.975
Global Motion Estimator	16.837±0.496	173.402±1.547	179.953±2.065
Patch Based Linear Regression	13.002±0.435	164.854±1.377	160.489±1.682
Patch Based CNN	11.480±0.431	105.242±0.839	101.880±1.042
Whole Image Dynamic Convolution Network	12.340±0.461	117.316±0.929	118.402±1.174
Patch Based Dynamic Convolution Network	11.114±0.412	101.983±0.802	98.790±0.995

表 1. 结果. 三个数据集上的竞争方法比较。

运动时，这个方法就会失败。

基于线性回归的块 一个线形回归模型，转换 4 个 70×70 雷达图片成 10×10 的序列中下一个图片块的中心块。该线形回归模型在训练集中被学习。序列中下一个图片 200×200 的中心区域在滑动窗口方法中运用线性回归模型计算。

基于卷积神经网络的块 CNN 用于转换一系列 4 个 70×70 雷达图片块成 10×10 的序列中下一个图片块的中心块。CNN 在训练集中被学习。序列中的下一个图片重复运用滑动窗口方法中的网络计算出来。CNN 的架构是这样的：4 个 70×70 雷达图片块被传给有 32 个 7×7×4 滤波器的传统卷积层(C1)。得到 32 个特征映射传递给最大池层(M1)，该层有最大超过 2×2 的步幅为 2 的空间块。紧随着有 32 个 5×5×32 滤波器的传统卷积层(C2)。产生的 32 个特征映射被传递给另一个最大池层(M2)，该层有最大超过 2×2 的步幅为 2 的空间块。接着有 32 个 3×3×32 滤波器的传统卷积层(C3)。产生的 32 个特征映射被传递完全连接层 F1，产生一个 1600 项的向量。最后，这 1600 项被传递给完全连接层 F2，产生一个 100 项的向量，它就是该网络的预测。

具体的网络结构在试验和误差方式中选择，以获得最佳性能上的验证集。我们测试了约 20 个结构，包括大的和小的，并选择了表现最好的网络架构用于比较。

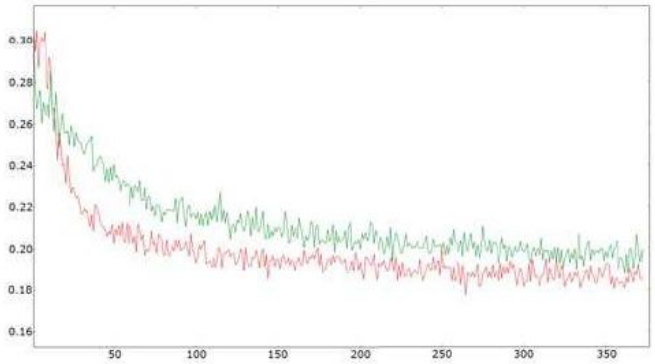

图 6. 在 Kansas 城市的验证集上的验证误差率作为一个时间的函数。X 轴是时间，Y 轴是验证集上的欧式损失。展示了基于 CNN 块(绿色)的误差和基于动态卷积层块(红色)的误差。

6.5 精度比较

对于每一个方法和每一个样本 t，欧式损失(平方)的计算公式：

$$\sum_{i=1}^{200}\sum_{j=1}^{200}(\hat{y}_{ij}^{t}-y_{ij}^{t})^{2} \tag{8}$$

$\hat{y}^{t}$ 代表预测图。意味着标准误差(SE)也在表 1 中被提出。所有在表中看见的不同都会被验证成在重要的 0.01 水平上的统计学重要事物，通过使用一对 t-测试，期望基于 CNN 块和基于动态卷积层块在 Tel Aviv 集上的不同。可以看出，Tel Aviv 集有着更小的误差。这是由于覆盖以色列的云雨层的稀少，相比于两个美国城市。在方法中，基于动态 CNN 的块在三个数据集中国提供最小误差率。接着表现最好的方法是传统 CNN 的块，然后

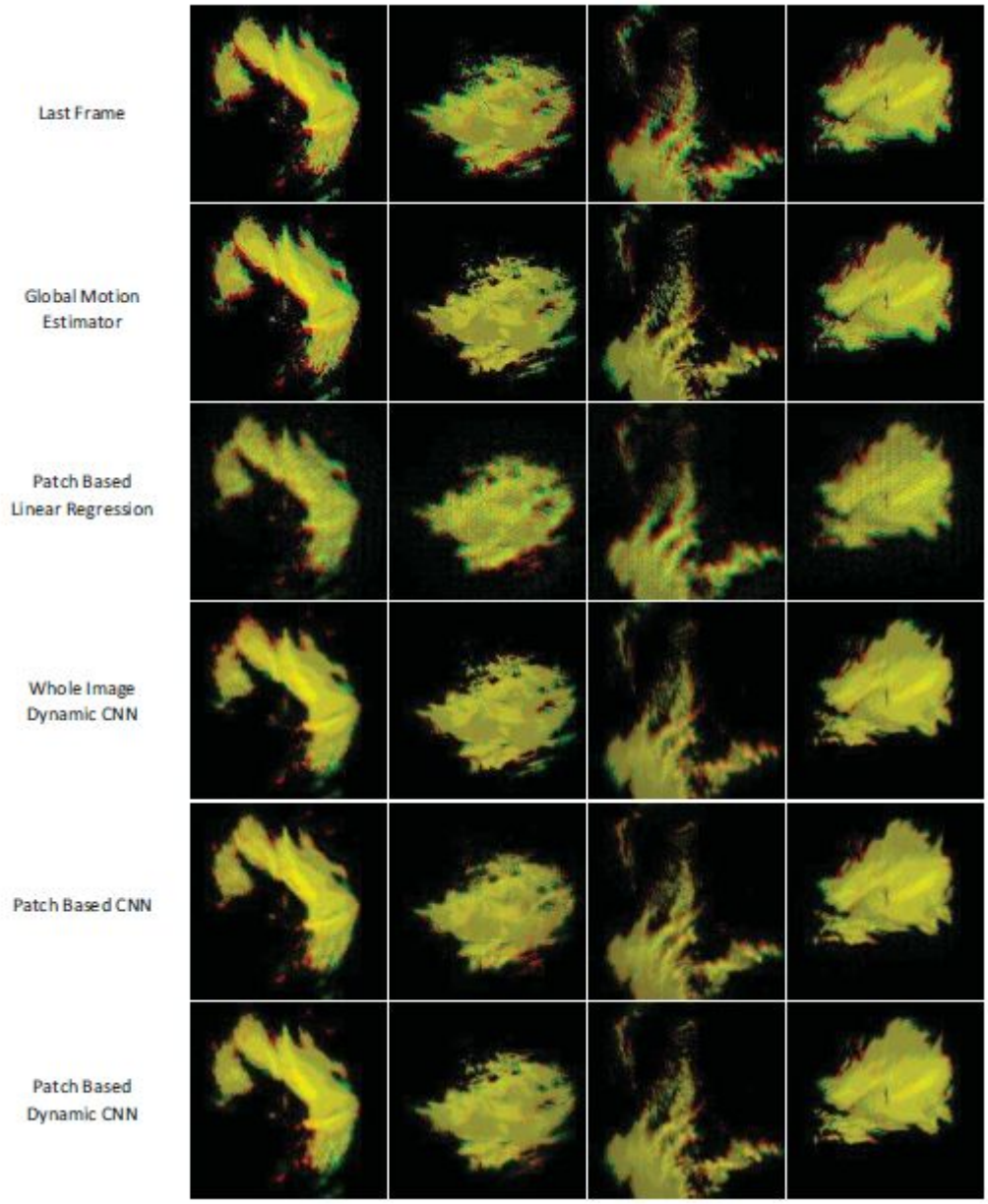

图 7.每一行代表一种方法，每一列表示同一序列。在每一张图片中，红色表示地面实况-序列中下一张真实雷达图片，绿色表示根据具体方法的预测，黄色表示预测和地面实况重叠部分。

表现最好的方法是全图动态 CNN。线形回归和全局运动估计表现并不如基于 CNN 的方法好。

补充，因为 Tel Aviv 数据集上缺少云雨层，我们计算 Tel Aviv 数据集在 US 数据集中学习的基于 CNN 的块和基于动态 CNN 的块的表现。结果，基于动态 CNN 的块欧式损失提高到 10.766±0.414，基于 CNN 的快的欧式损失降低到 11.708±0.536。

动态 CNN 方法的集合跟随着经典模式。网络的误差率开始比类似的传统 CNN 更高，但是一小段时间后，误差率迅速下降，并比相应的低。图 6 中描述了一个典型的运行图。总体而言，基于动态 CNN 块的集合比基于传统 CNN 块更快。为了

提供一个平等的集合比较，每一个时期的运行时间将会为基于动态 CNN 的块和基于 CNN 的块计算，通过采用 31 个时期的平均数。基于 CNN 的块的一个时期的运行时间是 189.935±3.950，基于动态 CNN 的块的一个时期的运行时间是 233.677±10.331。因此基于动态 CNN 的块的一个时期比基于 CNN 的块慢一点。但是基于动态 CNN 的块的总体集合比基于 CNN 的块快。

雷达图片序列例子和每一种方法预测的下一张图片展示在图 7。

7. 讨论

最近深度学习文献的重点是多类分类。然而深度学习网络能用于超分辨率[10]，图像盲反卷积[23]，噪声去除，和其他输入是图像的图像处理任务。这样的域可以受益于合成基于输入图像构成的动态滤波器。

另外，深度学习系统联合识别与检测和分割，如语义分割线的工作[2,8]也可以受益于动态滤波器。这种滤波器将允许系统适应多种场景和处理更广泛的场景。

虽然没有直接关系到动态卷积层本身，我们传播的误差信号源于欧氏误差。对于图像合成应用来说，这是不理想的，因为它不考虑图像结构。类如，模糊的图像，这没有视觉吸引力，有时比自然看清得到更低错误率，但是转移了图像。一个可能的方法来解决这个问题是通过在学习卷积神经网络的顶部合成图形化模型，等[12]。

最后，我们表明，包含一个动态卷积层的网络优于不包含的网络。因为动态层为计算滤波器包含子网，整体的网络变得更复杂。为了公平，我们试着一个基线网络，比被提的网络更深，并已经测试了多个代替品，学习率，以及规范化方案以获得最佳的表现的基线网络。

组成动态滤波器的被提及的网络更容易获得。更复杂的网络组件的使用时可能的，如动态卷积层，最终将导致简单的网络体系结构。

感谢

这项研究部分是由 Microsoft Azure Research Award 提供。

相关文献

[1] D. Alain and S. Olivier. Gated autoencoders with tied input weights. In S. Dasgupta and D. Mcallester, editors, Proceedings of the 30th International Conference on Machine Learning (ICML-13), volume 28, pages 154 - 162. JMLR Workshop and Conference Proceedings, May 2013.

[2] J. M. Alvarez, Y. LeCun, T. Gevers, and A. M. Lopez.Semantic road segmentation via multi-scale ensembles of learned features. In Proceedings of the 12th International Conference on Computer Vision - Volume 2, ECCV'12,pages 586 - 595, Berlin, Heidelberg, 2012. Springer-Verlag.

[3] L. Besson, C. Boudjabi, O. Caumont, and J. Parent du Chatelet. Links between weather phenomena and characteristics of refractivity measured by precipitation radar. Boundary-Layer Meteorology, 143(1):77 - 95, 2012.

[4] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

[5] D. C. Cires,an, A. Giusti, L. M. Gambardella, and J. Schmidhuber.Mitosis detection in breast cancer histology images

with deep neural networks. In Medical Image Computing and Computer-Assisted ntervention –MICCAI 2013, pages 411–418. Springer, 2013.

[6] N. Cohen and A. Shashua. Simnets: A generalization of convolutional networks. arXiv preprint arXiv:1410.0781, 2014.

[7] C. G. Collier. Flash flood forecasting: What are the limits of predictability? Quarterly Journal of the Royal Meteorological Society, 133(622):3–23, 2007.

[8] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Toward real-time indoor semantic segmentation using depth information. JMLR, 2014.

[9] A. Dehghan, E. Ortiz, R. Villegas, and M. Shah. Who do i look like? determining parent-offspring resemblance via gated autoencoders. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1757–1764,June 2014.

[10] C. Dong, C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In D. Fleet,T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision ECCV 2014, volume 8692 of Lecture Notes in Computer Science, pages 184–199. Springer International Publishing, 2014.

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1915–1929, 2013.

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, August 2013.

[13] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In International Conference on Artificial Intelligence and Statistics, pages 249–256, 2010.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[17] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013.

[18] R. Memisevic. Learning to relate images: Mapping units, complex cells and simultaneous eigenspaces. CoRR, pages –1–1, 2011.

[19] R. Memisevic and G. Hinton. Unsupervised learning of image transformations. In Computer Vision and

Pattern Recognition, 2007. CVPR ’07. IEEE Conference on, pages 1－8, June 2007.

[20] R. Memisevic, C. Zach, G. Hinton, and M. Pollefeys. Gated softmax classification. Advances in Neural Information Processing Systems, 23:1－9, 2010.

[21] D. E. Rumelhart, G. E. Hinton, and J. L. McClelland. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter A General Framework for Parallel Distributed Processing, pages 45－76. MIT Press, Cambridge, MA, USA, 1986.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Wilson. Learning representations by back-propagating errors. pages 533－536, 1986.

[23] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Sch¨olkopf. Learning to deblur. arXiv preprint arXiv:1406.7444, 2014.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.

[26] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1701－1708. IEEE, 2014.

[27] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In Proceedings of the 11th European Conference on Computer Vision: Part VI, ECCV’10, pages 140－153, Berlin, Heidelberg, 2010. Springer-Verlag.

[28] T. Weckwerth, C. Pettet, F. Fabry, S. Park, M. LeMone, and J. Wilson. Radar refractivity retrieval: Validation and application to short-term forecasting. Journal of Applied Meteorology, 44(3), 2005.