# # Image Generation from Scene Graphs#

Input:Scene graph

1. 经过graph convolution network，输入是每个obejct和他们之间关系的embedding vectors

2. 得到object和他们之间的关系然后去预测bounding box 和segmentation mask，组合成scene layout

3. 通过cascaded refinement network产生image

4. 为了保证图像的真实性，再训练一个GAN网络

具体操作：

## Scene Graphs：

模型的输入是scene graph，会给出点集合以及边集合，点集合描述了object,边集合描述了他们之间的关系

使用训练好的embedding layer 将边与点转化为dense vector.

## Graph Convolution Network:

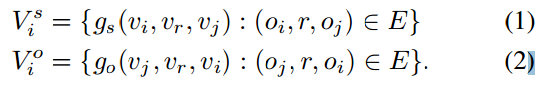
输入是点和边的vectors，通过函数计算产生一个新的向量，函数只计算相邻的输入，所以信息会沿着边集合传递

输入Vi,Vj属于object o,edges(oi,r,oj)属于 edges set，同过三个函数gs,gp,go来计算，

输出：Vi',Vj',Vr'

Vr' = gp(Vi,Vr,Vj)

计算Vi'比较复杂，因为需要收集所有与oi相关的边与点：

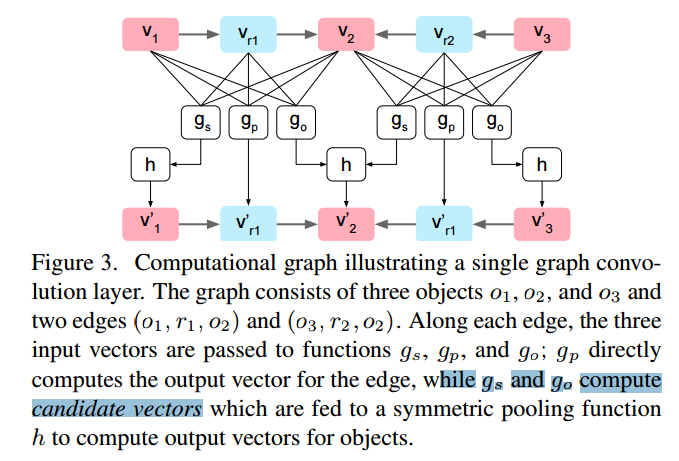


（1）表示以Oi开始的边对应的节点与边

（2）表示以Oi结束的边对应的节点与边

Vi’ = h((1)U(2))

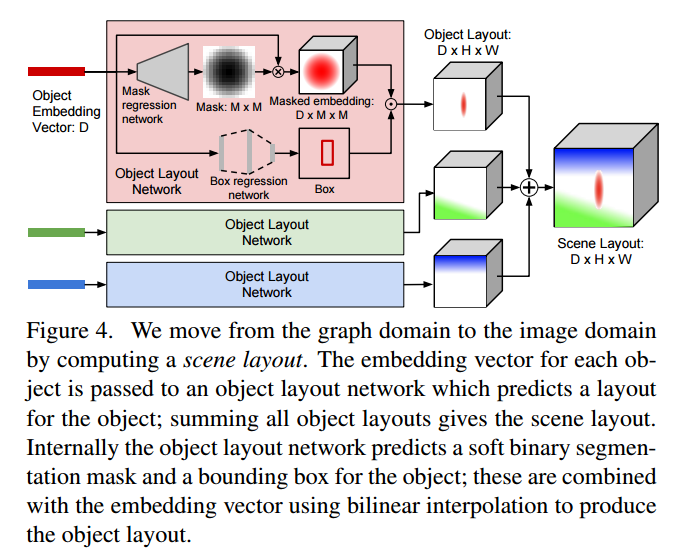
gs,gp,go是一个MLP，有三个输入，三个输出。h 平均其输入向量然后输入到一个MLP



## Scene Layout(将graph domian转换到image domian):

通过对每个物体预测一个bounding box 和segmentation mask来建立Scene Layout。

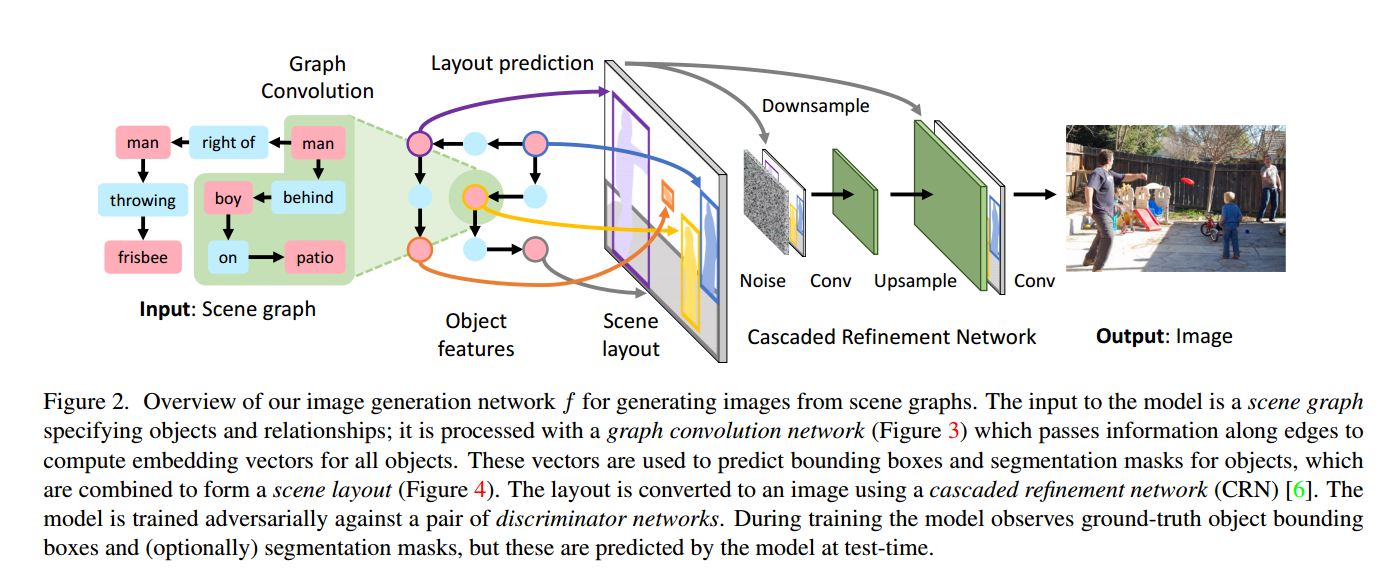
输入一个embedding vector（D维），通过mask regression neteork预测一个二进制的mask（MxM）,通过box regression network预测一个boundung box,将embedding vector与mask相乘得到masked embedding（DxMxM），然后通过双线性插值将其warp到bounding box的位置，这样得到一个object layout,将多个layout累加起来就得到了Scene Layout。



## Cascaded Refinement Network：

给定了Scene layout，所以必须产生于其对应的图像。

输入：一个Scene layout和从上一个module中的输出。这两个输入在通道上进行级联，然后通过一系列的卷积操作，然后使用最近邻插值。第一个module使用高斯噪声作为输入。



损失函数：需要优化六个损失函数：

Box loss =



Mask loss = 预测值与GT逐像素的交叉熵

Pixel loss =



Image adversarial loss :Gan损失函数，使图片看起来更真实

Object adversarial loss :Gan损失函数，使物体看起来真实

Auxiliarly classifier loss：要使生成的物体分类能分到该类中

具体网络结构见论文附录