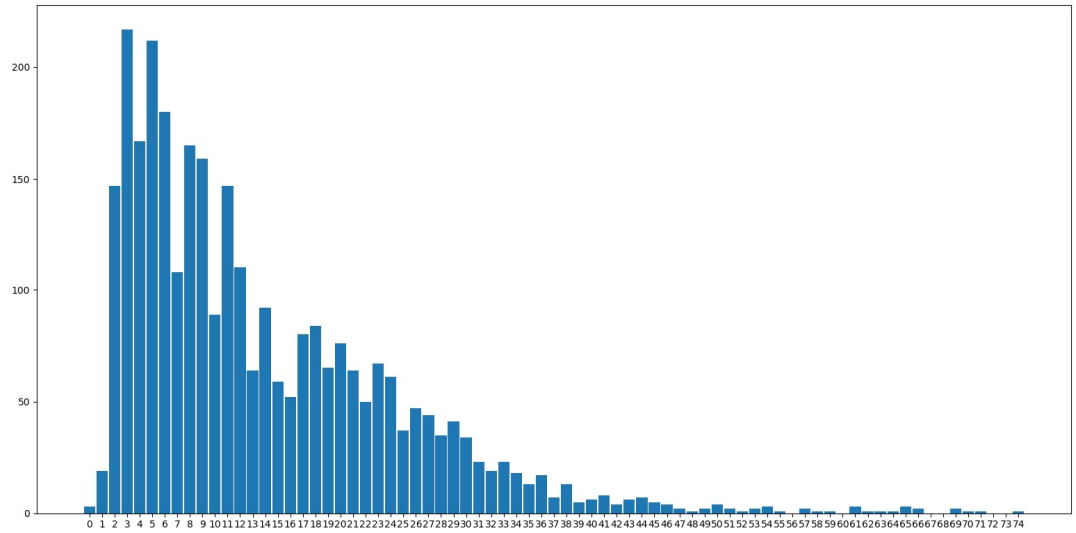
本周把网络按照 autofocus layer 那篇论文的思路进行了修改，把 resnet3D50 的 layer3 的最后一个 bottleneck 改成了多个平行的 dilation rate 不同的卷积结构。这个模块的输入是一个 1024*(L/8)*7*7 的 feature map X，通过.detach()得到一个复制 tensor，这个 tensor 通过 3*3*3 的卷积将 channel 先将为 channel/2，后面接一个 relu 激活函数，再通过 1*1*1 的卷积将 channel 变为 branch 的个数个，这个权重为了和多个 branch 相乘需要进行 dim=1 的 softmax。将 X 通过不同 dilation rate 的卷积和 bn 层得到的 feature map 与权重的卷积进行相乘，每个权重的 feature map 的 channel 为 1，所以需要进行 expand，相乘后的 feature map 的 size 与输入 feature map 相同。重复这样的模块两次，在最终的 relu 前加上 X，形成一个残差结构。

大概统计了一下训练集的 ground truth 之间的帧长，如下图所示



横坐标值为帧长除以 8 求整的值，纵坐标表示不同长度的 ground truth 的个数，超过横坐标最高值 74 的还有 16 段 ground truth，也存在上千帧的视频，暂时我把 branch number 取为 4，用 3*3*3 的核进行卷积，dilation rate 取值为(1,1,1),(4,1,1),(8,1,1),(16,1,1)，即在视频长度的维度上进行取空洞卷积，同时通过 padding 来确保输出的 feature map 的 size 不发上变化。

现在网络还没训练出来，我想看看两个 autofocus 的模块中间的 channel 从 1024 改为 256 减少计算量对网络带来的影响，还有一方面就是 dilation rate 取的是否合理，以及多个模块加入对准确度的影响。学长能不能详细讲一下求导算感受野的方法，这个没有搞明白。目前在把 tridentNet 的结构移到自己的网络中来，也在看学长给的那篇文章。主要想先把感受野计算这块搞明白，然后再尝试把两边的思路合并。