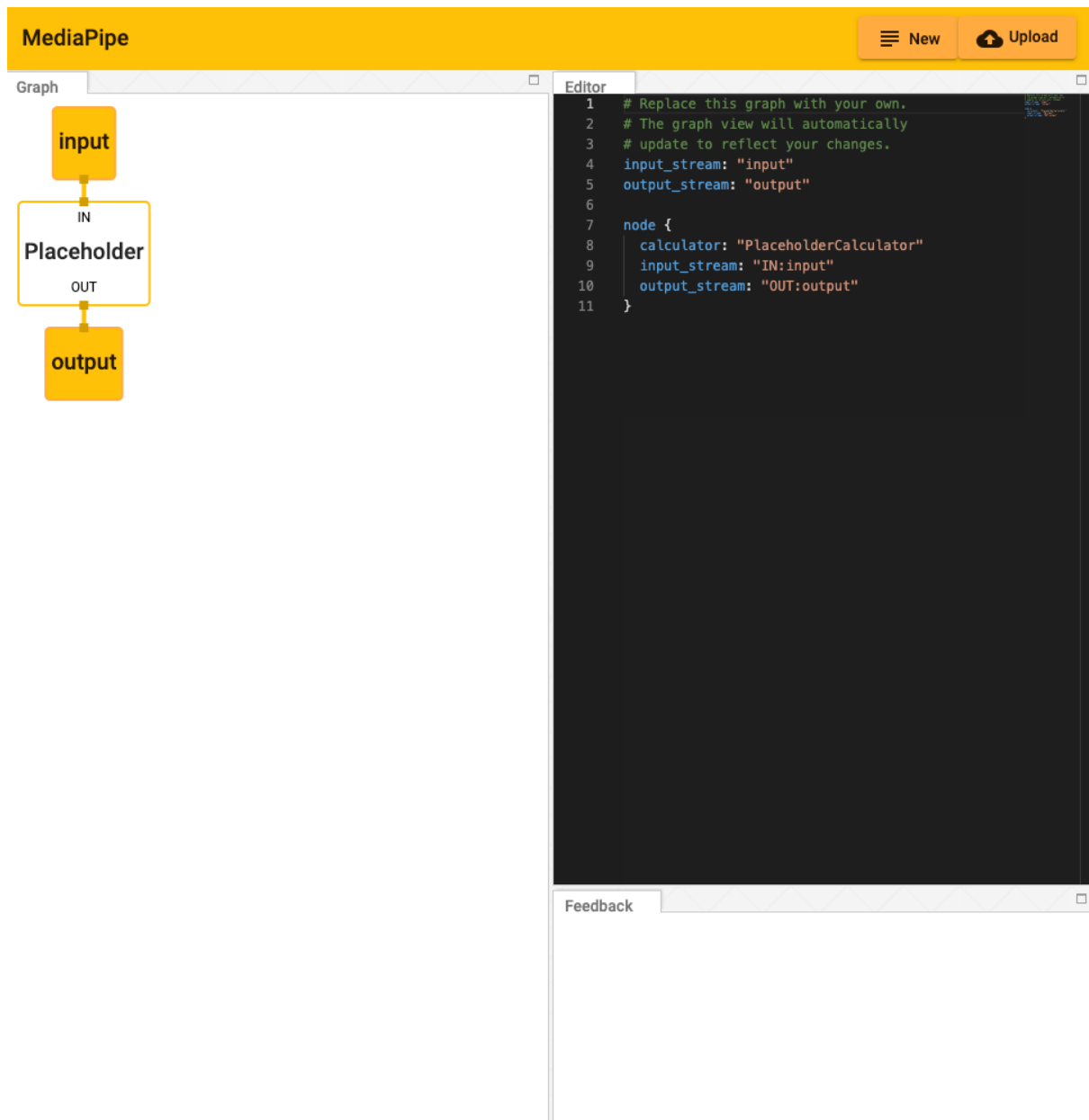


## AI学习笔记--MediaPipe--Graphs Visualizer

为了便于理解图形处理的Calculator结构，并理解机器学习推理的整体行为，MediaPipe构建了一个MediaPipe Visualizing 的可视化工具。

地址：<https://viz.mediapipe.dev/>

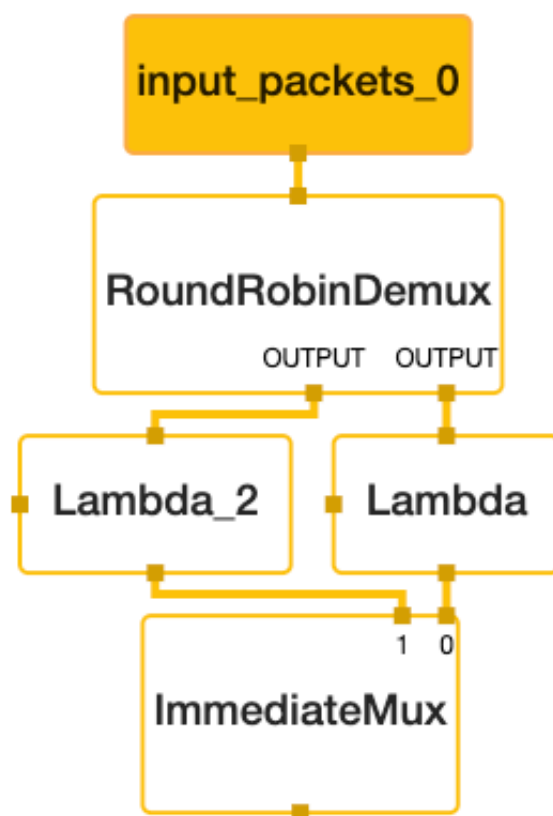
- 图形视图工具允许用户通过图形配置来查看Calculator 的关联。也可以修改图形编辑器或上传。用户可以可视化并排除他们创建的Graph中出现的逻辑异常。



可以通过在编辑器视图添加和编辑代码来修改，最后输出到左侧图形。

```
Editor
1  # Replace this graph with your own.
2  # The graph view will automatically
3  # update to reflect your changes.
4  input_stream: "input"
5  output_stream: "output"
6
7  node {
8      calculator: "PlaceholderCalculator"
9      input_stream: "IN:input"
10     output_stream: "OUT:output"
11 }
```

可以点击编辑，编辑相应的 Graph 信息，也可以直接上传 Ptxt 文件到上面，用于查看，还能够 copy/parse 的方式把代码 push 到 view 当中。可视化工具显示了各个 Calculator 之间的连接关系。流输出从上一级 Calculator 输出到下一级 Calculator 顶部，并且每个输出和输入流对应的 type 也有表示。



```
input_stream: "input_packets_0"
node {
    calculator: "RoundRobinDemuxCalculator"
    input_stream: "input_packets_0"
```

```

    output_stream: "OUTPUT:0:input_0"
    output_stream: "OUTPUT:1:input_1"
  }
  node {
    calculator: "LambdaCalculator"
    input_side_packet: "callback_0"
    input_stream: "input_0"
    output_stream: "output_0"
  }
  node {
    calculator: "LambdaCalculator"
    input_side_packet: "callback_1"
    input_stream: "input_1"
    output_stream: "output_1"
  }
  node {
    calculator: "ImmediateMuxCalculator"
    input_stream_handler {
      input_stream_handler: "ImmediateInputStreamHandler"
    }
  }
  input_stream: "output_0"
  input_stream: "output_1"
  output_stream: "output_packets_0"

```

如果逻辑 OK，最后可以输出保存到 ptxt 文件中。用于整个 mediapipe 的 build 过程。