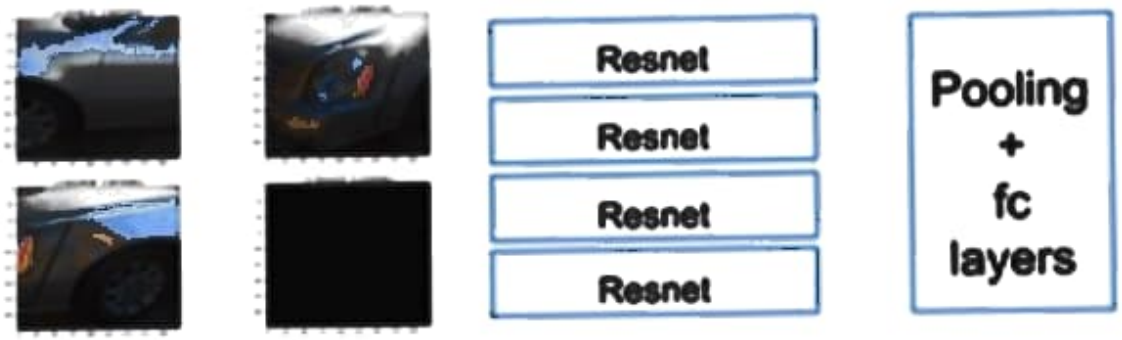
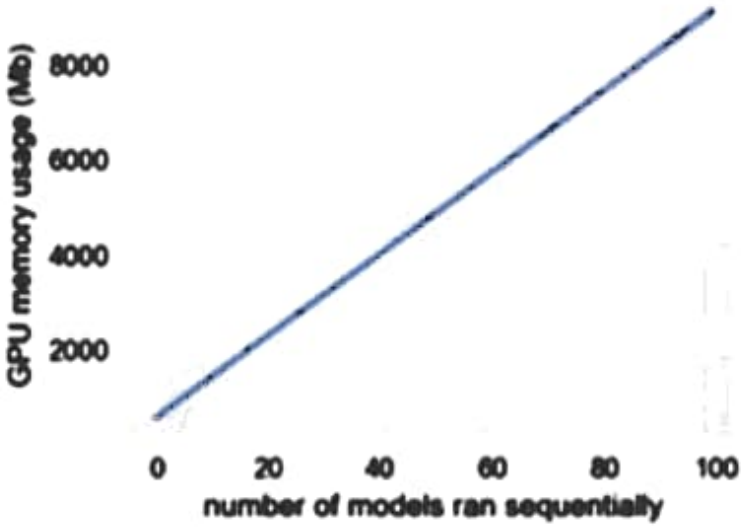


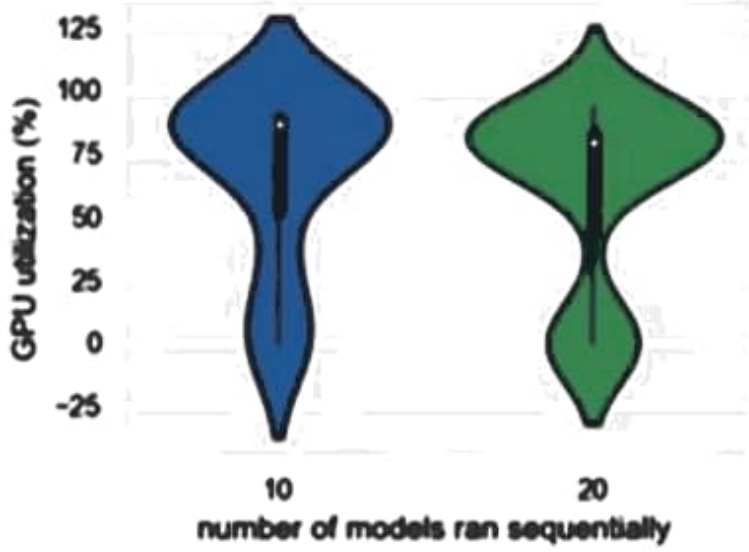
added this Python snippet: MIL v1 ▾

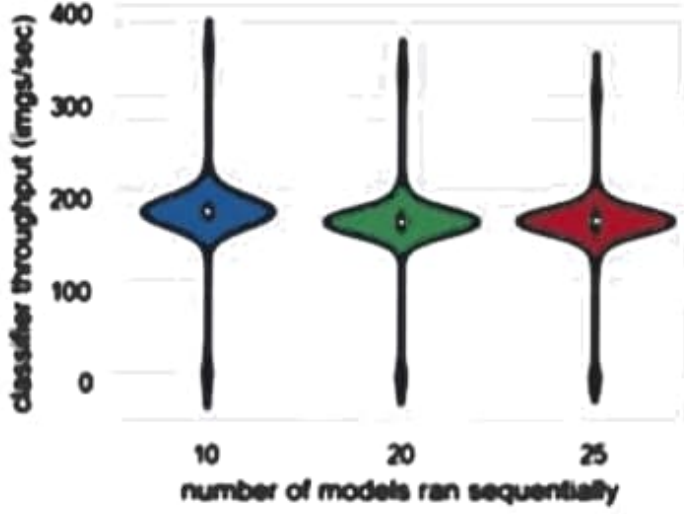
```
model = resnet.build_model(architecture, layer_count)
model = pop_layer(model)

im_per_claim = 4
mil_input  = Input(im_per_claim, 224, 224, 3)

ftr0 = model(Lambda(lambda x: x[0,:,:,:], output_shape=(224,224,3))(mil_input))
ftr1 = model(Lambda(lambda x: x[1,:,:,:], output_shape=(224,224,3))(mil_input))
ftr2 = model(Lambda(lambda x: x[2,:,:,:], output_shape=(224,224,3))(mil_input))
ftr3 = model(Lambda(lambda x: x[3,:,:,:], output_shape=(224,224,3))(mil_input))

ftr_all = merge([ftr0, ftr1, ftr2, ftr3], mode='concat')
fc1 = Dense(name='mil_fc1', activation='relu')(Flatten()(ftr_all))
```

Strategy 2: Results



- Increased the number of models runnable on same GPU to 80 on the same GPU
- Sequential inference still provides better GPU utilization





Recognise Parts → Assess Damage → Predict Operation → Map to £££

Visual models — Metadata models