# CS236605 Project Report

Jonas Brami 932162530
Ilay Chen 313432031

## Human Pose Estimation

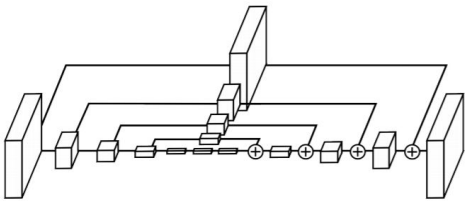
## Abstract

Our project goal is to improve the current model of 2D Human Pose detection which is defined as the problem of localization of human joints in 2D images.
In this project, we will mainly focus on the latest state of the art implementations of the HourGlass modules (HG) and its building block, the HourGlass residual unit (HRU).(https://arxiv.org/pdf/1603.06937.pdf)
We will implement and use the ideas of these papers with changes of our own and compare the results we obtained with theirs. (https://arxiv.org/pdf/1702.07432.pdf)
Finally, we will demonstrate how a simple change in the HRU leads to an increase of its receptive field which will lead to the overall improvement of our model accuracy, in addition to that - we'll present our idea for a simple change in the HG architecture that led for an extra improvement (inspired by https://arxiv.org/pdf/1804.06208.pdf).

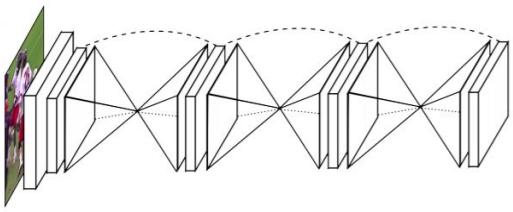
## Introduction

Our work is based on a CNN architecture called "stacked hourglass", designed for the task of human pose estimation. (https://arxiv.org/pdf/1603.06937.pdf)
The stacked HG performs a repeated bottom-up and top-down processing with intermediate supervision, it is motivated by the need to capture information at every scale - many cues are best recognized at different scales in the image. local evidence is essential for identifying features like faces and hands, while a final pose estimate requires global context. The network is based on the successive steps of pooling and upsampling that are done to produce a final set of predictions, it uses skip connections to preserve spatial information at each resolution.
Each layer of the HG consists of a residual module which contains convolution layers that uses filters no bigger than 3x3, down-sampling\up-sampling is performed between each residual module.

Visualization of a single HG:



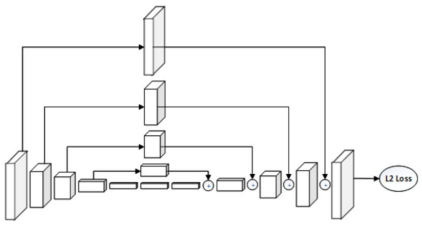
Visualization of the stacked HG:

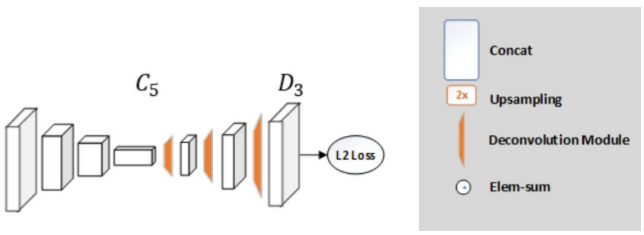


In the left image, each block is a residual unit consisting of 3*3 and 1*1 convolutions, downsampling (first half) and upsampling. Notice that resulted heatmaps are being produced at the end of each HG, where the intermediate supervision is being performed as well - which means that the predictions of each hourglass in the stack are supervised, and not only the final hourglass predictions. Stacking multiple HourGlass allows each HourGlass to use the information gathered by the preceding HourGlasses. Since the position of the joints are dependant, the detection of a specific limb by one HG will help the following HourGlass to affine this detection and to detect more parts of the body. The improvements we implemented on this stacked HG model are inspired by 2 other papers: *Simple Baselines for Human Pose Estimation and Tracking* and *Multi-Context Attention for Human Pose Estimation.*
These changes were made at 2 different levels of the original model: The first is an improvement  increasing the HRU receptive field and the second is the use of deconvolution instead of upsampling to densify the features gathered at each resolution.

The *Multi-Context Attention for Human Pose Estimation*:
In this paper, the authors propose a few mechanisms to improve the stacked HG model results, we focused on one proposed idea. Before finding this paper we both agreed that using the HG's residual unit as it is doesn't produce the maximal results as it's potential. We noticed that per residual unit the stacked HG computes the results only on one scale - we wondered what would happen if we'll enlarge its receptive field so each residual unit would have a better spatial understanding of each area.
The paper suggests a new idea to implement in the residual unit of the original stacked HourGlass paper, the authors called it Hourglass Residual units (HRUs).
The original stacked HG residual unit consists of three convolution layers and a skip layer, the paper offers to add another branch in order to have a larger receptive field, we'll discuss it in the "method" paragraph.

*Simple Baselines for Human Pose Estimation and Tracking:*
This work is introducing the use of deconvolution modules over the last convolution stage of ResNet which led to an increase in performance. This change was motivated by the need to generate high resolution feature map at each deconvolution.

1. 


2. 

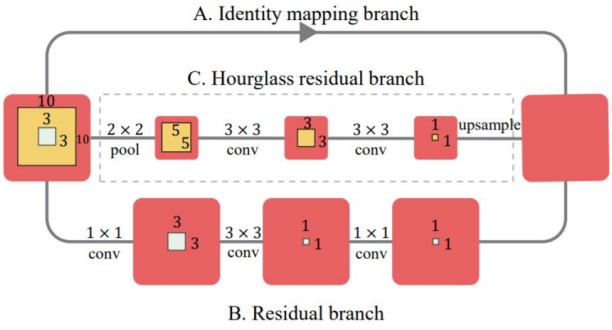
While the illustration 1. Is representing one HG module, the 2. Is representing the last convolution stage of ResNet called C5 where deconvolution modules are added to increase the heatmap resolution.

In our work, we decided to take the idea of using transposed convolution to increase the feature maps resolution and use it in the bottom up part of the HourGlass module instead of the original upsampling layer.
This is motivated by the fact that each transposed convolution module contains a learnable kernel whereas the original stacked HourGlass paper used simple nearest neighbor upsampling module (which doesn't contain any learnable parameters).

In our model, we use translated convolution as a learnable way of upsampling so that at each upsampling layer, features detected by each HRU of the HourGlass module are refined and the model is able to achieve fine-grained control over the increase in resolution.

**Methods**
We've used an existing code of the stacked HourGlass network and we modified it as cued before, we'll discuss it below:
Regards the HRU (Hourglass Residual Unit) modification - the stacked HG code had the original residual unit, so we had to create a similar class with wanted extensions:
The new branch that the paper suggests consists of max pooling, two convolution layers and upsampling.

Visualization of the new residual unit:




A and B are the original branches, C (the middle branch) is the new one

we've added Convs, Batch-Norm and ReLU layers to the class as expected and we changed the forwarding routine so the paper's theory will get involved.

Let us explain how does the receptive field changes - for the branch A, the receptive field is 1, for branch B we use 3x3 kernel at most, so the receptive field is 3. The new branch (C)  uses 2x2 max-pooling and two 3x3 convolution kernels which corresponds to 10x10 region of the input.

With regard to the upsampling in the bottom-up part of the HourGlass module, the original implementation of the stacked HourGlass was using the interpolate function with a scale factor of 2.
Instead of this, we used a translated convolution using 3 by 3 kernel to limit the increase of learnable parameters and make the learning process faster.

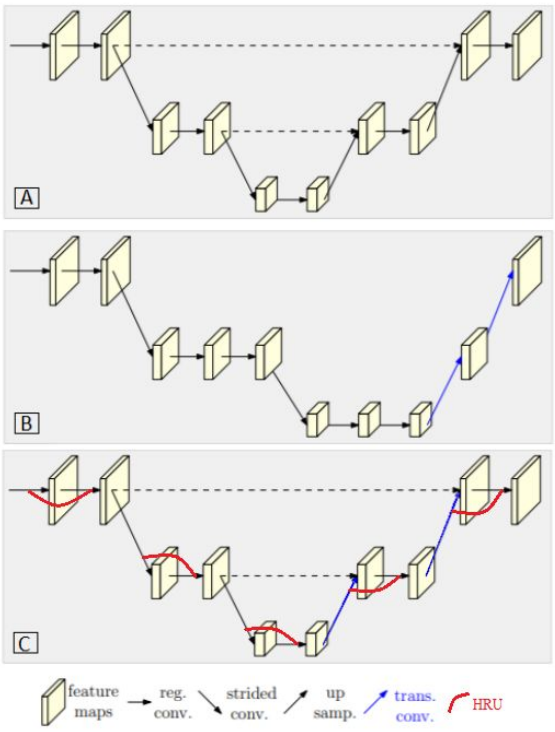
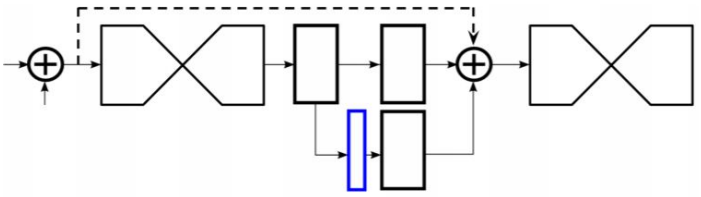


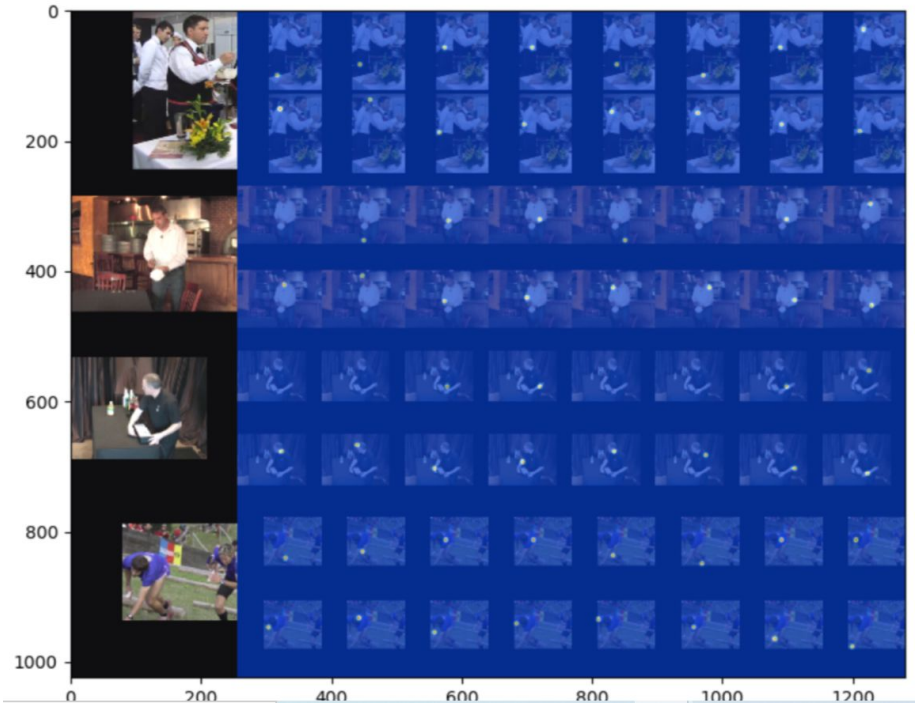

Illustration A is representing the original HourGlass module implementation.
Illustration B is including the changes proposed in the paper *Simple Baselines for Human Pose Estimation and Tracking*
Illustration C is representing our change, where the upsampling units have been replaced by deconvolutions.

### Implementation and experiments

First, we will describe in more depth the output of the HourGlass module and then we will talk about the experiments done on it.



This illustration is showing how the HG module are connected between each other and how intermediate supervision is done. In this picture, each rectangle is a residual module and the little blue rectangle represents the set of output heatmaps where we can apply loss.




This illustration is showing the output of the heatmaps. As we can see, for each image we want to find the pose, we get a set of heatmaps where each heatmap identifies a specific body part from the list below -
(right and left) ankles, knees, hips, wrists, elbows and shoulders.
Pelvis, thorax, upper neck and head top.

We've created two new models: (1) the stacked HG with the new residual unit. (2) the stacked HG with the new residual unit and deconvolution layers for up-sampling.
We performed several experiments on each model we've created - we trained the models on a variety of number of stacks - {1, 2, 8}.
Finally, we compared the validation accuracy of the models on each stack's number.
Our evaluation metric is similar to the one presented in the stacked HG paper which is the standard PCK (Percentage of Correct Keypoints) metric, it reports the percentage of detections that fall within a normalized distance of the ground truth.

All of our experiments were done using the MPII dataset consisting of 20k annotated images.
Since the network has an input resolution of 256x256, and running the HG at this resolution would require a too large amount of memory, we decided to use HG modules with an input resolution of 64x64.
To accommodate for this resolution and bring the 256x256 images to 64x64, our networks starts with a 7x7 convolutional layer with stride 2 following by a residual module and max pooling.

Data augmentation includes rotation (∓30 degrees) and scaling (0.75-1.25). As expected, the labels of the 'ground truth' changes in the same distortion.
The loss metric includes a Mean Squared Error, it is applied comparing the predicted heatmap to the 'ground truth' heatmap consisting of a 2D gaussian (with std of 1 px) centered on the body part location.

The optimization method used is RMSProp with a learning rate of 0.25 (inspired by Dr. Michael Zibulevsky from CS236330😃).

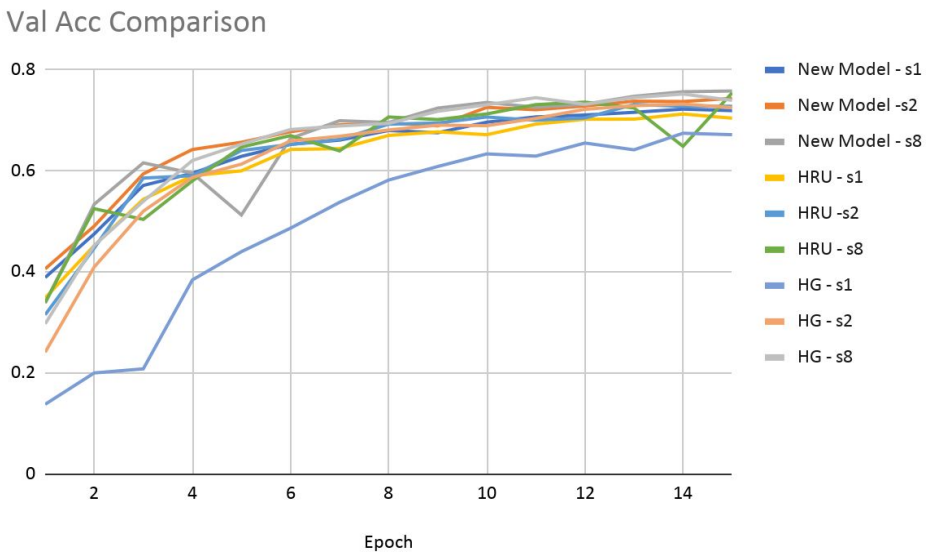
## Results

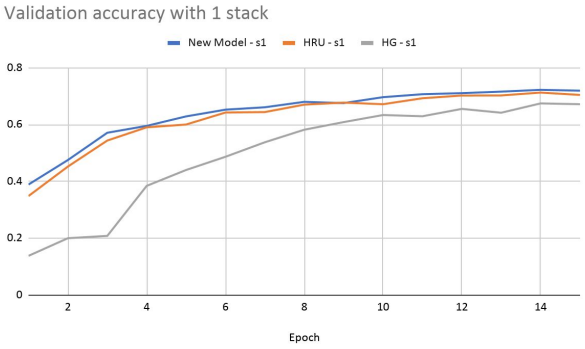
Bellow is the table of accuracies for each model we discussed previously:

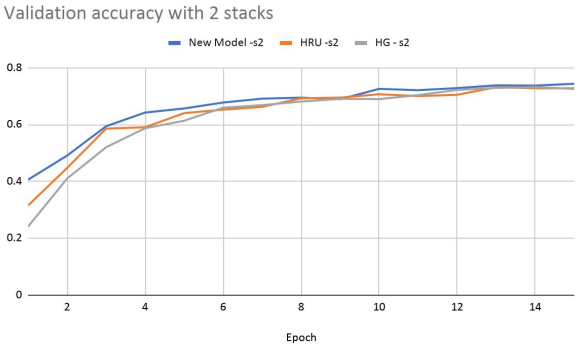
- HG is the original HourGlass model.
- HRU is the original hourGlass where the residual unit has been implemented as explained before.
- New model is the original hourGlass model where the HRU unit has been implemented as explained in the last paragraphs (HRU design) and where we used deconv layers (transconv2d) instead of upsampling with Nearest Neighbor.

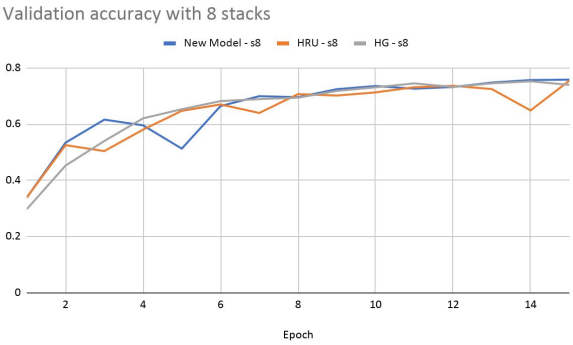
Notice that - s# represent the number of stacks the model uses.

| Epoch | New Model - s1 | New Model -s2 | New Model - s8 | HRU - s1 | HRU -s2 | HRU - s8 | HG - s1 | HG - s2 | HG - s8 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.38948 | 0.406531 | 0.338933 | 0.348974 | 0.31569 | 0.33976 | 0.138562 | 0.241599 | 0.297853 |
| 2 | 0.475693 | 0.491246 | 0.534437 | 0.452734 | 0.447608 | 0.525611 | 0.200655 | 0.41029 | 0.452979 |
| 3 | 0.571613 | 0.594517 | 0.616265 | 0.544243 | 0.586202 | 0.504458 | 0.208725 | 0.520655 | 0.540218 |
| 4 | 0.595745 | 0.642677 | 0.59598 | 0.590706 | 0.590905 | 0.580648 | 0.384714 | 0.58755 | 0.620946 |
| 5 | 0.629188 | 0.657075 | 0.513216 | 0.600548 | 0.640735 | 0.647578 | 0.44063 | 0.613809 | 0.653317 |
| 6 | 0.652894 | 0.677941 | 0.664649 | 0.642824 | 0.652855 | 0.670261 | 0.487058 | 0.659768 | 0.682254 |
| 7 | 0.661274 | 0.69155 | 0.699699 | 0.644427 | 0.662769 | 0.639969 | 0.538204 | 0.668926 | 0.689263 |
| 8 | 0.680348 | 0.695064 | 0.696157 | 0.670589 | 0.692887 | 0.707092 | 0.582253 | 0.681404 | 0.694374 |
| 9 | 0.675889 | 0.689544 | 0.72463 | 0.677657 | 0.695163 | 0.701965 | 0.609111 | 0.690809 | 0.718351 |
| 10 | 0.696791 | 0.726059 | 0.735812 | 0.67204 | 0.707048 | 0.713185 | 0.634015 | 0.689921 | 0.73123 |
| 11 | 0.707223 | 0.721407 | 0.726315 | 0.692967 | 0.700597 | 0.731531 | 0.629678 | 0.704233 | 0.745227 |
| 12 | 0.710976 | 0.728852 | 0.732328 | 0.702572 | 0.705097 | 0.736519 | 0.655569 | 0.722294 | 0.732268 |
| 13 | 0.716311 | 0.738376 | 0.748022 | 0.702855 | 0.731948 | 0.724998 | 0.64212 | 0.729348 | 0.745414 |
| 14 | 0.722653 | 0.737659 | 0.757041 | 0.712973 | 0.728162 | 0.649011 | 0.67491 | 0.732566 | 0.762462 |
| 15 | 0.719667 | 0.743941 | 0.758511 | 0.704691 | 0.728242 | 0.754759 | 0.672073 | 0.725707 | 0.740037 |






Before we compare the accuracies after 15 epochs, we will describe here the size of each model (in Bytes), the differences are due to the new residual design and the layers added:

| New Model - s1 | New Model -s2 | New Model - s8 | HRU - s1 | HRU -s2 | HRU - s8 | HG - s1 | HG - s2 | HG - s8 |
|---|---|---|---|---|---|---|---|---|
| 123,135,261 | 233,532,891 | 895,918,683 | 104,219,105 | 195,701,861 | 744,594,569 | 28,839,563 | 54,113,024 | 205,756,298 |

Now let’s compare the results -
As we can see in regards to the 1 stack experiment, both HRU and our models exceed the HourGlass original model accuracy results after 15 epochs. We couldn’t train every model for more than 15 epochs, but notice the constant tendency.

In regards to the 2 stacks experiment, we can see that all of the models accuracy converge approximately to the same value- but still, our model exceed the original HourGlass model and the HRU model by ~2%.

With regard to the 8 stack experiments, both HRU and our new model exceed slightly the accuracy of the original HourGlass model by 1.5%. Nevertheless, our model exceed the HRU accuracy only by 0.5%.