```
\documentclass{article}
\usepackage{caption}
\usepackage{booktabs}
\usepackage{graphicx}
\usepackage{subcaption}
\usepackage{listings}
\usepackage{xcolor}

\definecolor{RoyalBlue}{cmyk}{1, 0.50, 0, 0}

\lstset{
    keywordstyle=\color{RoyalBlue},
    basicstyle=\scriptsize\ttfamily,
    commentstyle=\ttfamily\itshape\color{gray},
    stringstyle=\ttfamily,
    showstringspaces=false,
    breaklines=true,
    frameround=ffff,
    frame=single,
    rulecolor=\color{black}
}

\title{Detecting Human Body Parts and Building Skeleton Models
    using Deep Convolutional Neural Networks}
\date{2017}
\author{Iftimie Florentin Alexandru}

\begin{document}

    \maketitle
    \pagenumbering{gobble}
    \newpage
    \pagenumbering{arabic}

    \tableofcontents
    \newpage

    \section{Introduction}

    Recent advances in real-time object detection with FasterRCNN enabled
    researchers to add new features to the model and make it flexible to new tasks.
    One task that was explored by researchers at Facebook was to add a segmentation
    branch that generated the contour of the detected object, creating a new model
    called FastMaskRCNN. The goal of this project is to replicate the above results
    and add new tasks to the model by adding body parts segmentation and key point
    regression.
    The first task that was researched during this project was the key point
    regression. Key point regression means that the human body has multiple joints
    and we need to find the x,y coordinates of those points. The best model that was
    found was OpenPose that was running at 2.3 fps on a Nvidia 960M GPU.
    Later in this project, the body parts segmentation task was researched. Body
    parts segmentation consists in detecting every person in the image and for each
    person find the contour of each body part. We found that although there are
    papers published about this task, there was no implementation available on the
    Internet.
    Such tasks are important for multiple applications ranging from the automotive
    industry, medical industry, fashion applications and robotic rescuing operations.
    For this project the task that was mostly researched was the body parts
    segmentation task and the model that was selected for modification and training
    was MaskRCNN.
```

```
\section{Related Work}
In this section is discussed the current state of the art models for key point
regression, as there are no publicly available models for body parts
segmentation.
There are two approaches to analyzing the human body for these kind of tasks.
The first approach is the top-down architecture that consists in detecting all
persons in an image and for each person detect their key points or segment their
body parts. This approach is based on the Faster-RCNN model that includes
additional task-specific branches.
The second approach consists in detecting all body parts in an image and create
graphs with their connections or propose some heuristics for reconstructing the
human pose. Most of these models are based on the Fully Convolutional Neural
Network approach and for each body part there is generated a heat-map that
detects the positions of a certain part in the image.
\subsection{Public models performances}
The below table shows the performance of the publicly available implementation
of different models. The below results are tested using a laptop with Nvidia
960M GPU and i7 processor. As it can be observed there is no model on body parts
segmentation.
\begin{table}[h!]
    \centering
    \caption{Performance of publicly available models}
    \label{tab:table1}
    \hspace*{-2cm}
    \begin{tabular}{cccc}
        \toprule
        Model & GPU & CPU & Results\\
        \midrule
        OpenPose 6 stages  & 2.3 fps  & Does not run on CPU  & Very accurate
        keypoints\\
        OpenPose 4 stages  & 3.1 fps & Does not run on CPU & Less acurate
        keypoints\\
        DeepCut  & Out of memory & 200 s & Very accurate keypoints\\
        Human part discovery & 10 s & Not tested & Bad segmentation\\
        Pose Singleperson & 0.53 s & Not tested & Very accurate keypoints\\
        Pose Multiperson & 1.7 s & Not tested & Very accurate keypoints\\
        DeepCut-cnn & Out of memory & 2.4 s & Very accurate keypoints\\
        Convolutional pose machines & Not available & 10 minutes & Very acurate
        keypoints\\
        \bottomrule
    \end{tabular}
\end{table}
\subsection{Datasets}
Over the Internet there are a lot of public datasets that have key point
annotations, body parts segmentation or both of them, for images with single
instances or multiple instances. The following table contains the list of the
datasets and their specifications. Some of them are not yet public and their
specifications remained unfilled.
\begin{table}[h!]
    \centering
    \caption{Databases}
    \label{tab:table1}
    \hspace*{-4cm}
    \begin{tabular}{cccccccc}
        \toprule
        Name &  Kp & Seg body & Seg parts & Size & Multiperson & Singleperson &
        Parts/Keypoints\\
        \midrule
        COCO&Yes&Yes&No&45K&Yes&Yes&17 Kp\\
```

```
        FLIC &Yes&No&No&25K&Yes&Yes&11 UpperBody Kp\\
        Freiburg&No&No&Yes&215&No&Yes&14 P\\
        LSP&Yes&No&No&10K&No&Yes&14 Kp \\
        MPII&Yes&No&No&25K&Yes&Yes&14 Kp \\
        Parse&Yes&No&No&305&No&Yes&14 Kp\\
        Poseevaluator&Yes&No&No&6K&Yes&Yes&6 UpperBody Kp\\
        Pose in the wild dataset&Yes&No&No&800&Yes&Yes&8 Kp\\
        PASCAL VOC&No&Yes&Yes&20K&Yes&Yes&22 P\\
        Chalearn&No&No&Yes&8K&Yes&Yes&14 P\\
        MIT SceneParsing&No&Yes&No&22K&Yes&Yes&14 P\\
        ADE20K&No&Yes&Yes&1.1K&Yes&Yes&>6 P\\
        PoseTrack&Yes&?&?&20K&Yes&Yes&?\\
        J-HMDB&Yes&Yes&Yes&13K&No&Yes&15Kp/14P\\
        Penn-Action&Yes&No&No&>10K&No&Yes&13 Kp\\
        HumanParsing&No&Yes&Yes&10k&No&Yes&18 P\\
        Fashionista&Yes&Yes&Yes&685&No&Yes&14K 14 P\\
        VideoPose&Yes&No&No&1K&No&Yes&6 Kp (upper and lower arms)\\
        VGG (Youtube, BBC) &Yes &No&No&23K&No&Yes&7 P\\
        FYDPP/UYDP&Yes&No&No&?&No&Yes&14 K\\
        \bottomrule
    \end{tabular}
\end{table}
\newpage
For the JHMDB database it was utilized a very convenient tool for annotating the
images as depicted in Figure \ref{fig:tool}. Such a tool makes the annotation a
lot faster while maintaining the quality. In the figure below the user is able
to drag am modify naturally the mask of the body parts  and of the key points.
The selected datasets for training the model of this project remained Pascal
VOC, ADE20K, Chalearn and JHMDB. Pascal VOC contains 1K images with persons in
it with very well annotated body parts. ADE20K contains 1K images and is poorly
annotated and with a lot of mistakes. Chalearn contains 6K images with decent
annotation, but not very precise. Also it does not have a lot of variation in
the dataset as the samples are drawn from a few videos. JHMDB is very well
annotated with 30K images but its downside is that the images contains a single
instance of a person.
\begin{figure}
    \centering
    \includegraphics[width=6cm]{tool.jpg}
    \caption{Tool for body parts annotation}
    \label{fig:tool}
\end{figure}

The links for these datasets are found in the references
\cite{Chalearn:1,ADE:2,JHMDB:3,VOC:4,VOC5:5}. In order to download the Chalearn
dataset, the user must register and then download the data \cite{CHALEARN6:6}.

\section{The steps and description of this solution}
\subsection{Fisrt attempts}
Before I went to modify MaskRCNN my first approaches to this problem were in
modular steps such as understanding how Fully Convolutional Neural Networks
works and understand what the transpose convolution does and how to improve the
segmentation precision. For this step I implemented my own Fully Convolutional
Neural Network using VGG as the backbone and overfitted on a single image as in
figure \ref{fig:FCNE}.

\begin{figure}
    \centering
    \begin{subfigure}{.5\textwidth}
        \centering
        \includegraphics[width=.8\linewidth]{segmentation.jpg}
```

```
        \caption{Predicted Segmentation}
        \label{fig:predseg}
    \end{subfigure}%
    \begin{subfigure}{.5\textwidth}
        \centering
        \includegraphics[width=.8\linewidth]{inputImage.jpg}
        \caption{Input Image}
        \label{fig:inimage}
    \end{subfigure}
    \caption{Fully Convolutional Neural Network Excercise}
    \label{fig:FCNE}
\end{figure}

The next thing relevant to my problem was to adapt MaskRCNN to detect all body
parts in an image as in Figure \ref{fig:bodyparts}. The problem with this
approach is that the model does not know that those body parts belong to that
person. If more persons are in the image, the problem becomes much bigger. I
quickly discarded this approach as the problem of assembling each part to each
person was a task too difficult to resolve it in useful time. Thus I needed my
model to be able to detect all persons in the image and detect their body parts
from a single shot, without any other heuristics on top of the model.
\begin{figure}
    \centering
    \includegraphics[width=6cm]{bodyparts.jpg}
    \caption{MaskRCNN applied on independent body parts detection and
    segmentation}
    \label{fig:bodyparts}
\end{figure}

\subsection{Final approach}
MaskRCNN is a model of a neural network that is able to do object detection,
classification and segmentation. The ideea is to detect all persons in an image
and segment their body parts. This is a real-time model running at 200 fps on a
Titan X GPU.
MaskRCNN is an extension of FasterCNN. The latter had 2 branches. One for region
proposal and one for the classification. This extension has 3 branches. One is
for for object detection another is for classification and bounding box
regression and another one is for segmentation. The initial model was generating
a mask for the entire object and my task during this project was to modify the
network to detect only persons and for each person segment their body parts.
Another task that is to be researched is the key point regression task. The
following images in Figure \ref{fig:MaskRCNN} depict the results on the training
set.

\begin{figure}
    \centering
    \begin{subfigure}{.5\textwidth}
        \centering
        \includegraphics[width=.9\linewidth]{segmentation1.jpg}
        \caption{Example 1}
        \label{fig:Example1}
    \end{subfigure}%
    \begin{subfigure}{.5\textwidth}
        \centering
        \includegraphics[width=.9\linewidth]{segmentation2.jpg}
        \caption{Example 2}
        \label{fig:Example2}
    \end{subfigure}
    \begin{subfigure}{.5\textwidth}
        \centering
```

```
        \includegraphics[width=.9\linewidth]{segmentation3.jpg}
        \caption{Example 3}
        \label{fig:Example3}
    \end{subfigure}
    \caption{MaskRCNN on person detection and body parts segmentation}
    \label{fig:MaskRCNN}
\end{figure}

\subsection{Personal Modifications}
This section shows how I modified the original code from the GitHub repository
by Charles Shang.
Starting from the root directory in the file libs/nets /pyramid\_network.py line
287 and 349 I modified the pooled\_height and pooled\_width to 56 by 56 pixels.
At line 360 in the same file, the mask is upsampled by using transposed
convolution to 112 by 112 pixels. The mask at that point has the shape
[N,112,112,256] and a [1,1] convolution kernel is applied to the Tensor in order
to get [N,112,112,7] where 7 is the number of parts in the image (6 body parts
and a mask that contains the segmentation for the entire body). I also included
the entire segmentation of the body because it is easier to segment and better
annotated. When drawing each body parts is clipped using the mask of the body in
order to fit better inside the boundaries. The output of the mask is passed into
a sigmoid layer that will help transform the values into probabilities.
Also, in libs/nets/pyramid\_network.py line 566 the function mask\_encoder also
must be modified in order to generate the proper mask targets given the
proposals. The mask targets are also reshaped to 112x112 pixels. The actual
targets are generated inside libs/layers/mask.py at line 75.

Training the neural network was done in different ways. First I trained with the
Pascal VOC dataset that contained about 1000 images but the model over fitted
the dataset immediately. I also played around with the output of the mask in
order to increase the precision of the segmentation. A mask of 112x112 pixels
was better than a mask of 56x56 pixels, but one problem that appeared when the
network was training was that at a certain point, the network was proposing 120
regions for a single image and that caused an Out-of-memory exception because
for a single convolutional layer the tensor shape was [120,112,112,256] that
occupied 1.4 Gb of GPU memory. My first attempt to overcome this problem was a
multistage training approach. The idea was to train the network for person
detection only then include the segmentation branch. I thought that training
only for human detection would reduce the number of proposals and it did. One
problem that I found here was to restore the parameters of the network as the
checkpoint only contained the parameters for the bounding box regression and
classification branches, and the restorer threw exceptions if the parameters for
the newly included segmentation branch were not found in the checkpoint.

After I managed to restore the parameters and leave the segmentation branch
parameters initialized with Gaussian distributions I resumed the training. But
the network got itself into the same problem stated earlier. The gradients that
contained the loss for the segmentation affected the parameters for the region
proposal network and as the training was running, the RPN was proposing more
regions in order to stabilize itself. After I found that there was a flag for
the maximum number of Rois per image I changed that from 256 to 80 and the
training continued without problems.

One thing to be mentioned is that I first started training this model by over
fitting on a single image to test if it works.
Initially I modified only for human body parts detection but in that form I did
not know which part belonged to which person. So in order to get rid of
reconstruction algorithms or branch and cut algorithms, I decided to transform
the output of the mask branch as inspired after the Fully Convolutional Network
where at the last layer for visualizing the results, the argmax for each class.
```

```
\subsection{Understanding the implementation}
The graph of the model generated in Tensorboard was not very clear and it
contained a lot of variables that were not necessary to visualize. Thus, in
order to study and understand how the model works and is trained I looked over
the entire code and generated my own graph. The following parts describe how the
model works. This scheme is found in mask\_rcnn\_final.xml and can be opened and
edited in draw.io . Generating this scheme allowed me to better understand the
code in order to be able to modify it.

The first part is the ResNet part visualized with the layers at different scales
as in Figure \ref{fig:ResNetPart}. It has the role of feature extraction. It
takes as input an image with any size and the C2, C3, C4, C5 layers are used for
the Feature Pyramid Network construction detailed in the following pages.

\begin{figure}[h]
    \centering
    \includegraphics[width=4cm]{1.jpg}
    \caption{ResNet part}
    \label{fig:ResNetPart}
\end{figure}

From the above figure, the C5 layer is upsampled step by step and fused with the
intermediate layers of the ResNet as depicted in Figure \ref{fig:up}.

\begin{figure}[h]
    \centering
    \includegraphics[width=4cm]{2.png}
    \caption{Upsampling part}
    \label{fig:up}
\end{figure}

For each step of the upsampling process such as P5,P4,P3 and P2, a region
proposal network is attached in order to generate proposals for different scales
as in Figure \ref{fig:up}. The region proposal network generates the bounding
box, the classification scores and the anchors.

\begin{figure}[h]
    \centering
    \includegraphics[width=4cm]{3.png}
    \caption{RPN part}
    \label{fig:RPN}
\end{figure}

For the region proposal part there is attached a loss function that takes the
proposed regions and the objectness score of each region and the generated
bounding box targets and score targets as in Figure \ref{fig:RPNTargets}.

\begin{figure}[h]
    \centering
    \includegraphics[width=4cm]{4.png}
    \caption{RPN targets part}
    \label{fig:RPNTargets}
\end{figure}

The region proposals, classifications and anchors from each region proposal
network (Figure \ref{fig:concat}) are all concatenated in their respective
tensors as in Figure .

\begin{figure}[h]
```