Capsule Networks for 3D Pose Estimation in Computer Graphics

Kenia Picos[1], **Ulises Orozco-Rosas**[1], **Alfredo Cuesta-Infante**[2], **Antonio S. Montemayor**[2], **Juan J. Pantrigo**[2]

{kenia.picos, ulises.orozco}@cetys.mx, {alfredo.cuesta, antonio.sanz, juanjose.pantrigo}@urjc.es

[1]CETYS Universidad, *Centro de Excelencia en Innovación y Diseño* (CEID), El Lago, Tijuana, Baja California, México,
[2]Universidad Rey Juan Carlos, *Departamento de Informática*, C. Tulipán, S/N, Móstoles, Madrid, Spain

Introduction

Pose estimation is an important task for novel engineering applications, such as virtual and augmented reality (VR/AR), pose-based video games, object reconstruction, target tracking, driving assistance and recent sports analytics. Commonly, an efficient pose estimation system depends on the pose visualization given by a 3D configuration of location, orientation, and scaling parameters of the target. In this work we implement Capsule Networks to solve 3D pose estimation in computer graphics of rigid objects using a multi-GPU architecture.

Pose Estimation with Capsule Networks

Capsule Networks (CapsNet) are composed by a convolutional neural network (CNN) with structures, called capsules, for correcting spatial relationships of a target within a scene. CapsNets have the property of learning a whole entity by first recognizing its parts.

As shown in Fig. 1, the output of the CNN is the input of a capsule. The output of the capsule consists of the probability of encoded features and a vector set of instantiation parameters, which ensures the invariance to estimate pose under texture and deformations. In CapsNet, a prediction vector $\hat{\mathbf{u}}_{j|i}$ indicates how much a primary capsule i contributes to a class capsule j. An agreement between capsules is carried out with the product of a coupling coefficient c_{ij}. A weighted sum $s_{ij} = \sum_{i=1}^{N} c_{ij}\hat{\mathbf{u}}_{j|i}$ yields the candidates for a squash function $v_{ij} = (||s_{ij}||^2 s_{ij})/(1 + ||s_{ij}||^2||s_{ij}||)$

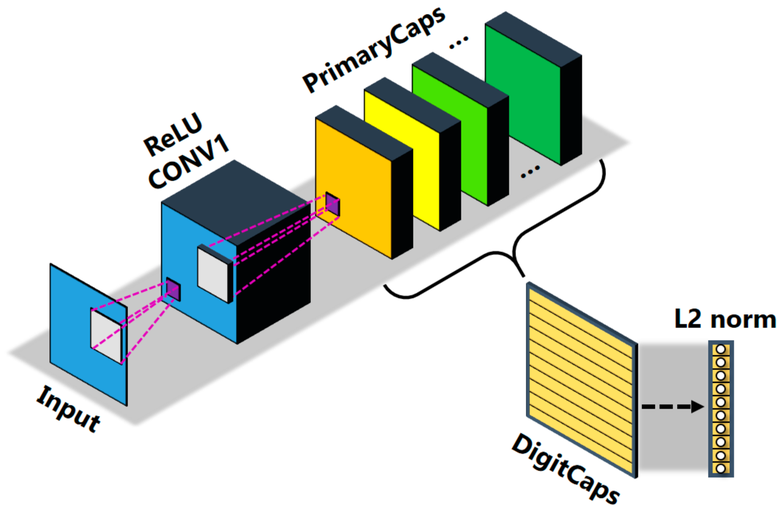

Figure 1: Basic architecture of a Capsule Network

Experimentation

In order to achieve pose estimation, the architecture implementation of the network consists on a Convolutional layer (256 channels, 9×9 filters, stride 1) with ReLU activation function; a Capsule layer (6×6×32 capsules, 9×9 filters, stride of 2) with squash function yields ten 16D capsules; a fully connected layer (DigitCaps) performs classification based on 10 classes. Image reconstruction is done using the decoder from fully connected layers.

We test with several pose configurations of a 3D model using OpenGL for graphics rendering. Training is performed on images of 28×28 pixels. The dataset consists in total of 70,000 images (each one corresponds to a pose configuration), which equals to 60,000 images for training and 10,000 images for testing. The experiments yield robustness to affine transformations of the target in terms of rotation angles and linear scaling.

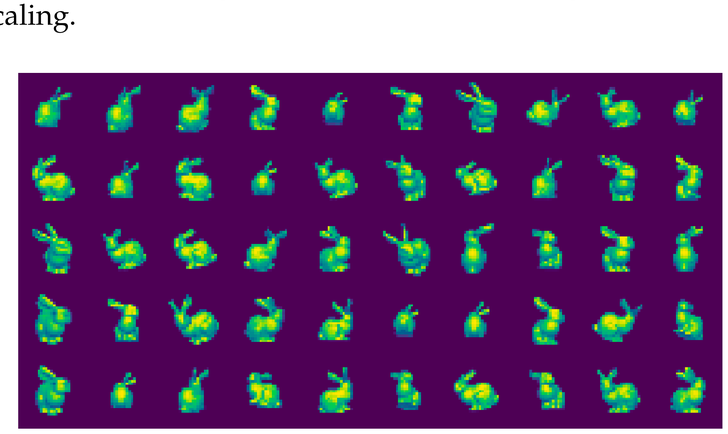

Figure 2: Original images from a digital model with different pose configurations

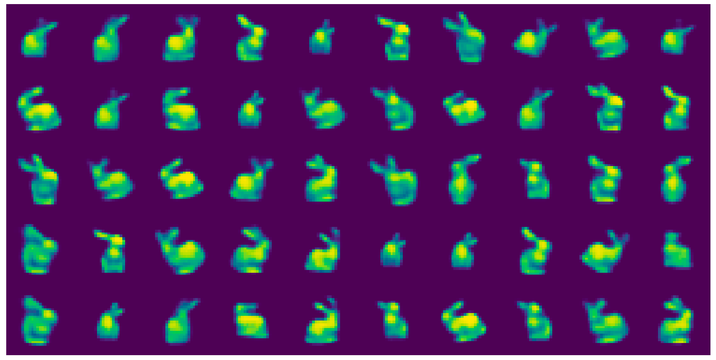

Figure 3: Reconstructed images of each pose configuration from the original images presented above

Performance Results

The experiments were achieved by using a CPU Intel(R) i9-9900K processor @ 3.60 GHz 16 GB RAM, and two NVIDIA graphics cards: GPU NVIDIA GeForce RTX 2080 Ti with 4352 CUDA cores @ 76T RTX-OPS, and a GPU GeForce GTX Titan Black with 2880 CUDA cores.

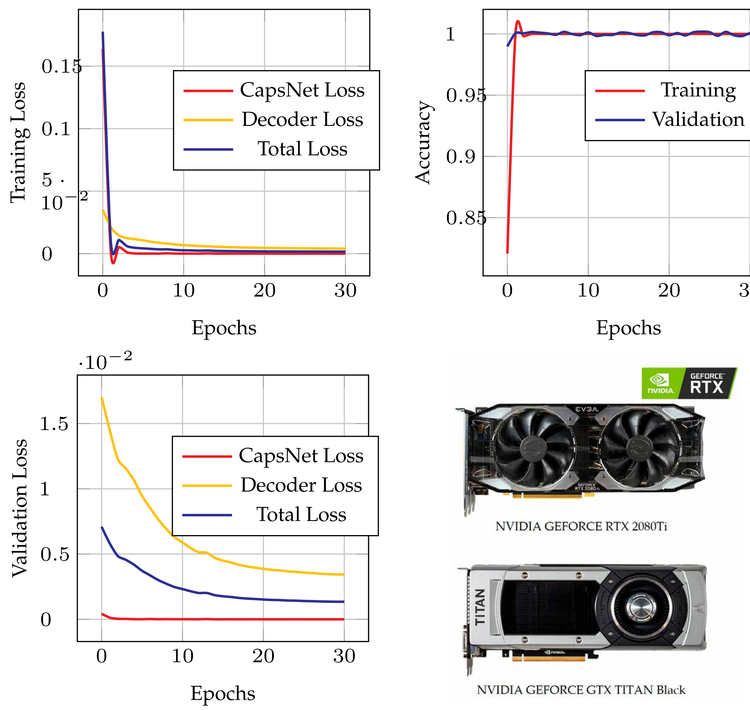

Figure 4: Performance results on a multi-GPU implementation

Conclusions

CapsNets are promising in terms of improving computer vision tasks for VR/AR applications. This work presents an efficient pose estimation of rigid objects using a multi-GPU architecture. The proposed implementation includes Capsule Networks yielding good experimental results in terms of training loss, accuracy, and 3D pose estimation performance.

Acknowledgements:

This research is supported by CETYS Universidad, Consejo Nacional de Ciencia y Tecnología (CONACYT), and Universidad Rey Juan Carlos.

References:

Sabour S., Frosst N., Hinton G., Dynamic Routing Between Capsules. NIPS 2017.

Hinton G., Sabour S., Frosst N., Matrix Capsules with EM Routing, Proc. ICLR 2018.

Picos K., Diaz-Ramirez V. H., Montemayor A. S., Pantrigo J. J., Kober V., Threedimensional pose tracking by image correlation and particle filtering, Opt. Eng. 57(7), 2018.