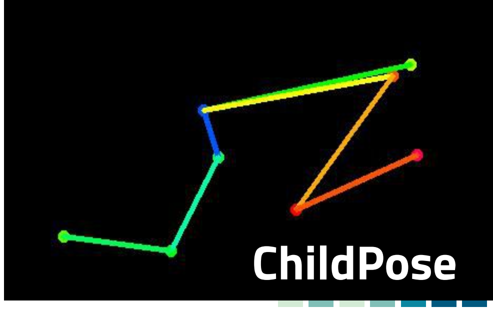
TOOTHTABLE

OUR APPROACH

Pose estimation using OpenPose: training on extracted poses

Transfer learning: Xception model

Ensemble model: combining prediction results



POSE ESTIMATION

- Pre-processing: use OpenPose to **extract keypoints** (joints, facial features), isolating human bodies from the background.

Model	Input	Accuracy
Fully-Connected	**Flattened keypoint coordinates**	**~45%**
CNN	**Stickman figure (as shown)**	**~40%**

- Problems: pictures may have missing keypoint information (hidden body parts), slow speed of pose extraction during evaluation, inability to determine if other people are interference or part of the pose (e.g. ChestBump, HandShake)

TRANSFER LEARNING

Why Transfer Learning?

- Small training dataset is insufficient for a well-trained CNN
- Limited time and GPU resources for training

What did we do?

- Use a pre-trained model for feature extraction capabilities, fine tune the model to our problem
- Initially, only last few layers of model were retrained
- Then, we tried unfreezing all layers and it gave us a better score
- Models tested: Xception, InceptionResNetV2, InceptionV3

Results?

- Test Set 1 (Xception): **~50+%**
- Test Set 2 (InceptionV3): **~70+% (final submission)**

OVERALL MODEL ARCHITECTURE

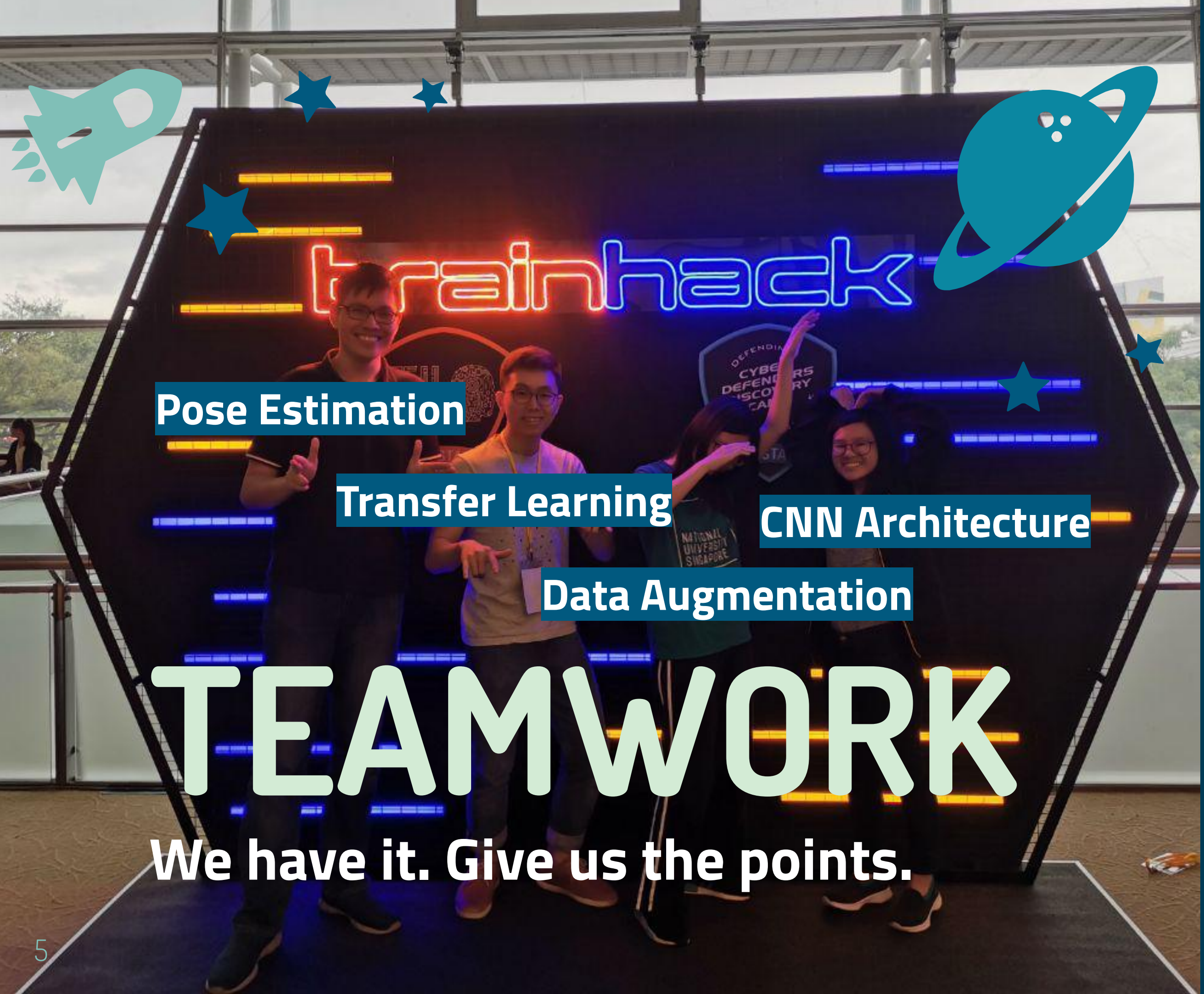
Data poses are taken at different angles, so training data and test data can look very different. **Over-fitting** is very likely.

Our solutions, given limited training data (the original 1322 images):

- Data Augmentation: increases data diversity
- Transfer Learning: capitalises existing datasets
- Regularisation: dropout layers, early stopping

Areas of improvement:

- Ensemble multiple models using weighted average
- Train a wider diversity of models (e.g. VGG, DenseNet)
- Add ReduceLROnPlateau callback



Pose Estimation

Transfer Learning

CNN Architecture

Data Augmentation

TEAMWORK

We have it. Give us the points.