Ship motion prediction with deep learning

Lance De Waele

GHENT
UNIVERSITY

Contents

GHENT UNIVERSITY

Introduction

- Royal Military Academy
- Robotics & Autonomous Systems lab
- Marsur
  - Developing an autonomous surface vessel (ASV)
- Marland
  - Developing an autonomous drone
  - Take off and landing on moving vessels

GHENT UNIVERSITY

Goal

- Predict the motion of the ASV
- Define optimal moment to land drone
  - Deck stabile and level
  - Moment with minimal impact
- Input (ZED mini sensor):
  - IMU live data
  - Stereo camera live video
- Output:
  - Sequence of Pitch, Roll and Heave
  - Up to 30 seconds ahead

GHENT UNIVERSITY

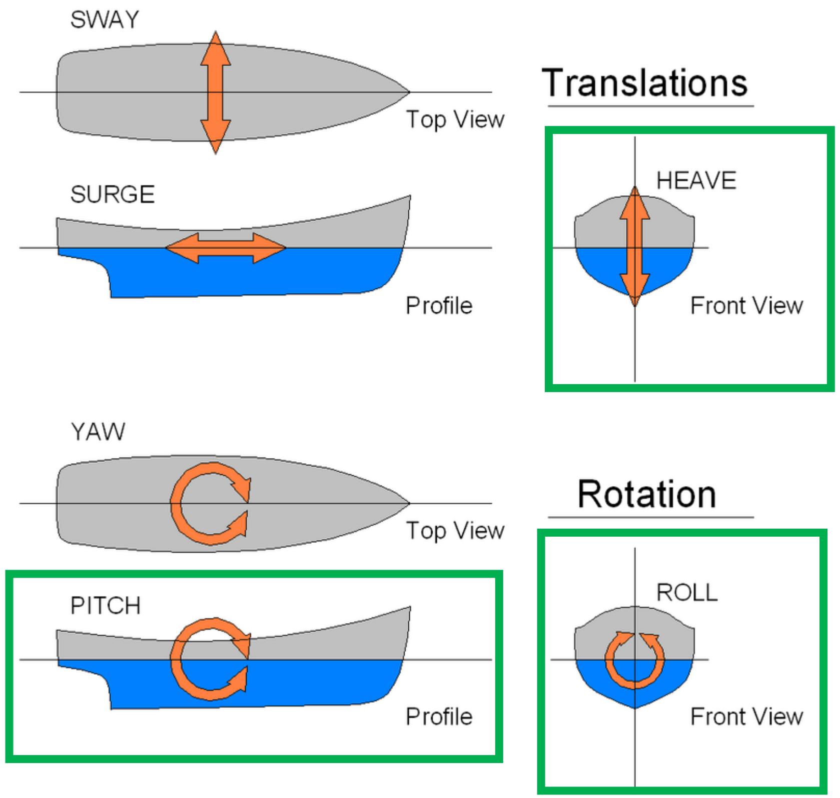
Implementation (1st semester)

- Evaluate the input and output data
- Test different existing deep learning models
- Define criteria for our case
  - Reference: best performing model
  - Baseline: worst performing model
  - Speed: real-time possible?

GHENT UNIVERSITY

Data

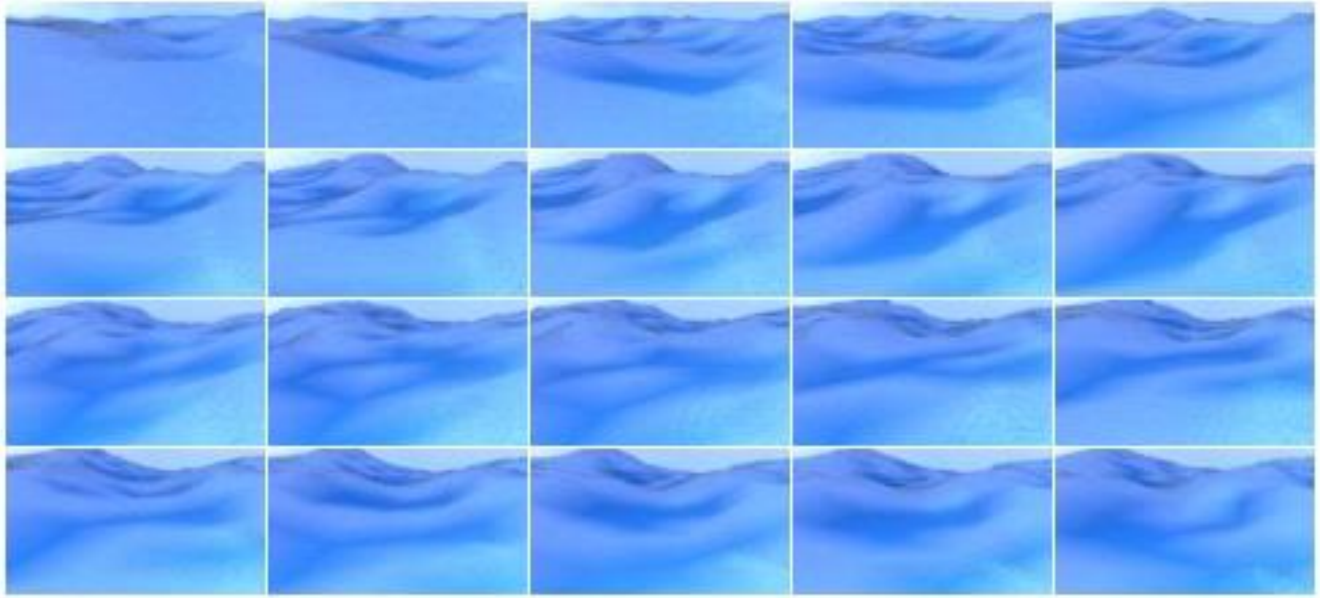
- Ship motion in 6 degrees of freedom
    - Translational
        - Sway
        - Surge
        - **Heave**
    - Rotational
        - Yaw
        - **Pitch**
        - **Roll**



GHENT UNIVERSITY

Data

- Synthetic data
  - Generated in a Blender[1] ship-wave simulation
  - 540 episodes of 400 images
  - Rendered at 2 frames/second
    - Avoid duplicate data at higher fps
    - 30 hours of simulation
  - Resolution 96x54
  - Pitch & roll per image
- Optional expansion
  - Data augmentation
    - Fog, contrast, …



GHENT UNIVERSITY

[1]Source: https://github.com/Nazotron1923/ship-ocean_simulation_BLENDER

Data

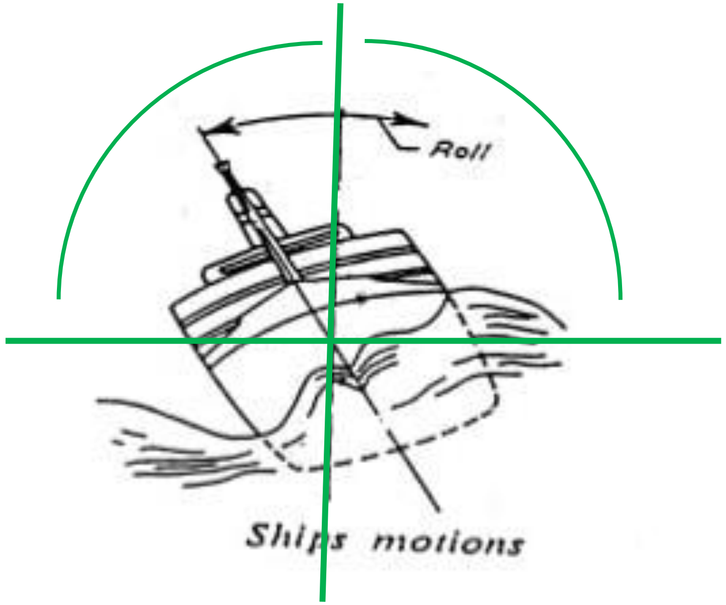
- Real data
  - Live video
  - Live data from IMU
  - Additionally
    - Wind sensor data
    - Thrust (ASV)
- Challenges
  - Find optimal fps
  - Objects in images
  - Stabilization (out of scope)

GHENT UNIVERSITY

Data Preprocessing

- Images need same size, scale and cropping
  - Already done in generated data set
- Pixel values
  - Normalize to [-1 , 1]: $p_{normalized} = \frac{p_{original} \cdot 2}{255} - 1$
- Pitch and Roll
  - Normalize angle with abs values: [-90°, 90°]
  - Normalize to [-1, 1]:

$$angle_{normalized} = \frac{angle_{original} - (-90)}{90 - (-90)} \cdot 2 - 1$$

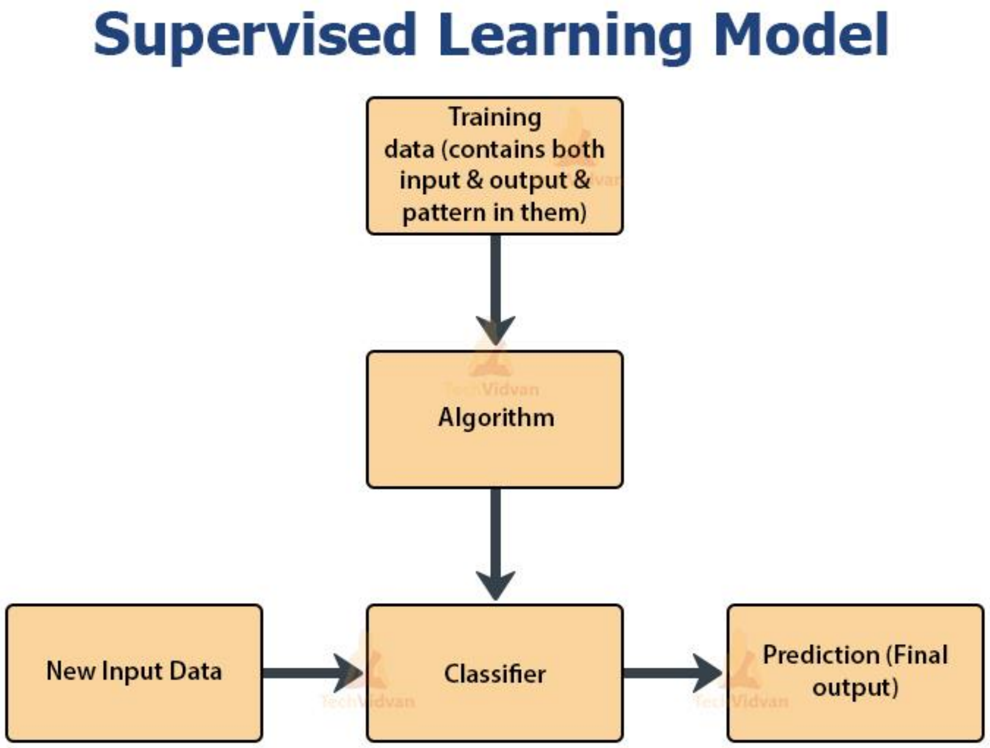


GHENT UNIVERSITY

Model training

- Supervised learning
- Maps an input to an output
  - Based on examples of input-output pairs

- Main components to perform training process



Model training - Parameters

- **Past window size**
  - Number of seconds of data that is taken to predict pitch and roll (sequence of frames)
- **Future window size**
  - Future window size - the number of seconds for which pitch and roll will be predicted (sequence of frames)
- **Number of episodes**
  - The number of episodes used as input data (max 540)
- **Latent vectors**
  - The dimensions of the latent vectors for the LSTM parts of the model

GHENT UNIVERSITY

Model training - Hyperparameters

- **Learning rate**
  - Determines to what extent newly acquired information overrides old information
- **Weight decay**
  - Adds a penalty to the error function
    - Depends on the magnitude of the weights that connect neurons to each other
  - Used to limit overfitting in a neural network
- **Batchsize**
  - Defines the number of samples to work through before updating the internal model parameters
- **Number of Epochs**
  - Defines the number of times that the learning algorithm will work through the entire training dataset

GHENT UNIVERSITY

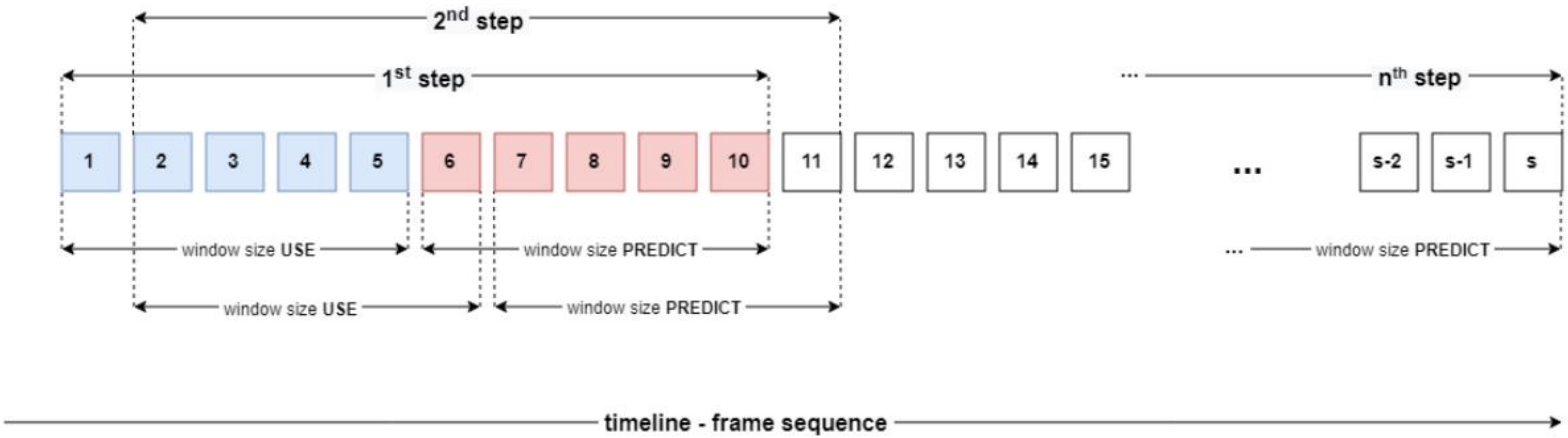
Model training – Data loader

- The input is a *sequence* of images
- Models **without LSTM** module
  - Length of the sequence = past window size

GHENT UNIVERSITY

Model training – Data loader

- Models **with LSTM** module
  - The length of the sequence = [60, 72, 120] frames
  - Based on used LTSM model
  - The passage through this sequence is carried out step by step



GHENT UNIVERSITY

Model training – Data split

- Generated data has increasing complexity
  - Wave height, wind speed, …
- All data sets need data of varying complexity
  - For best model performance
- Every 10% of data is split randomly in
  - Training data
  - Validating data
  - Test data

GHENT UNIVERSITY

Model training – Optimization

- Training speed optimizer
  - ADAM algorithm
- Early stopping
  - Prevent overfitting and memory resources
  - Ready-made solution from github[1]
- Hyberband
  - HyperBand algorithm: mix of grid search and random search
  - Only best model's parameters were individually optimized

GHENT UNIVERSITY

[1]Source:https://github.com/Bjarten/early-stopping-pytorch

Model Testing

Model testing – Loss function

- Regression problem -> Mean Square Error
- MSE = sum(distance(target, prediction))
  - Target: angle
  - Prediction: angle^

$$loss = MSE = \frac{\sum_{i=1}^{number frames}(angle_i - \hat{angle_i})^2}{number frames}$$

GHENT UNIVERSITY

Models testing - parameters

- Basic parameters for testing
- Hyperband
  - Only good models

Parameters	
Number episodes	540
Number epochs	50
Learning rate	1e-04
Weight decay	1e-03
Frame per second (fps)	2
Past window size	10 s
Future window size	12 s

GHENT UNIVERSITY

Proposed Models

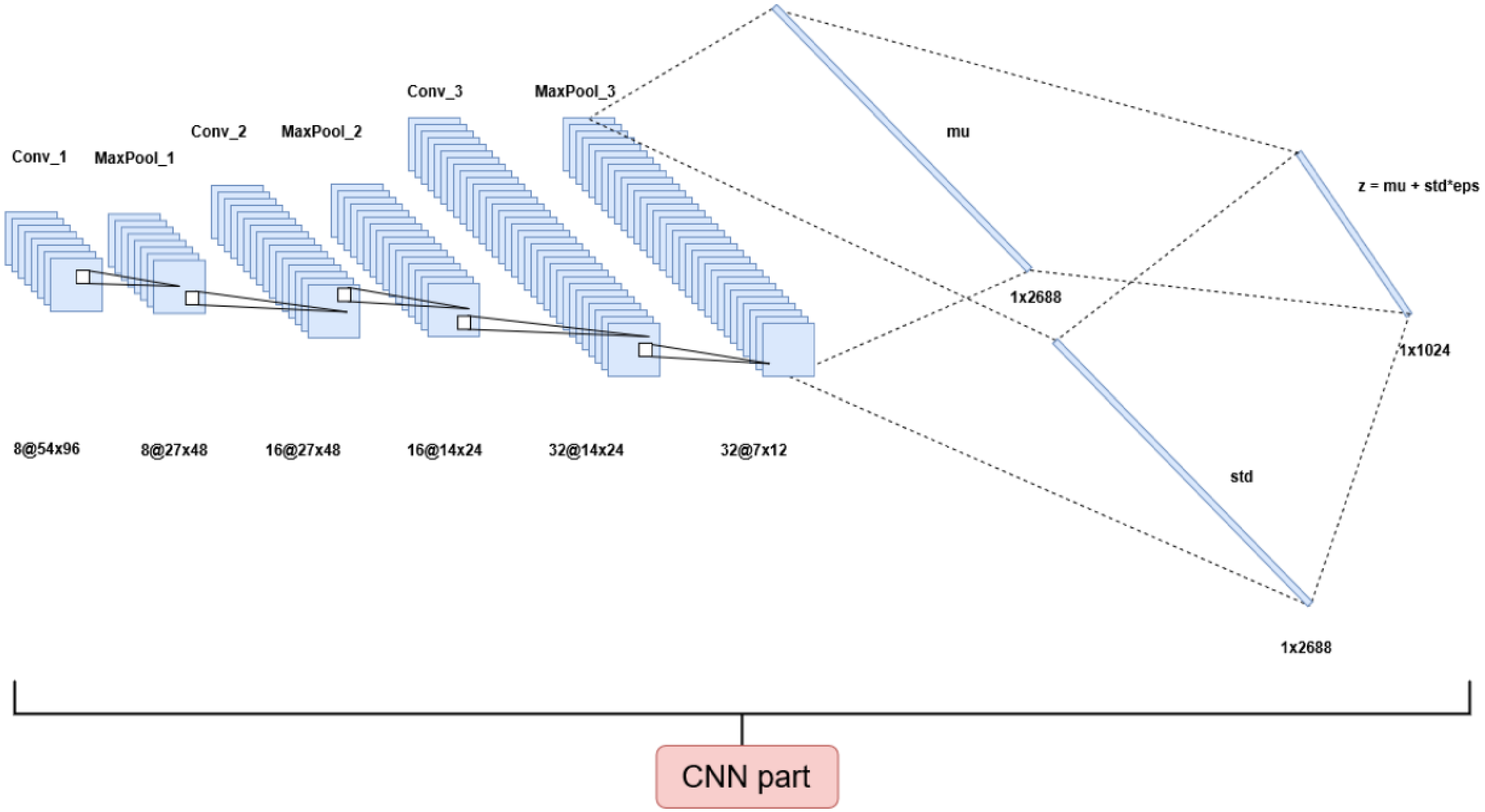
GHENT UNIVERSITY

Introduction – CNN, LSTM

- Convolutional Neural Networks - CNN
    - Analyse images and extract feature
- Long Short-Term Memory networks – LSTM
    - RNN capable of learning long-term dependencies
    - Process sequences of data
    - Suited for predicting time series data

Introduction – Autoencoder

- Extract features into vector
- Encoder part
  - Pre-trained component
- Greater perfomance

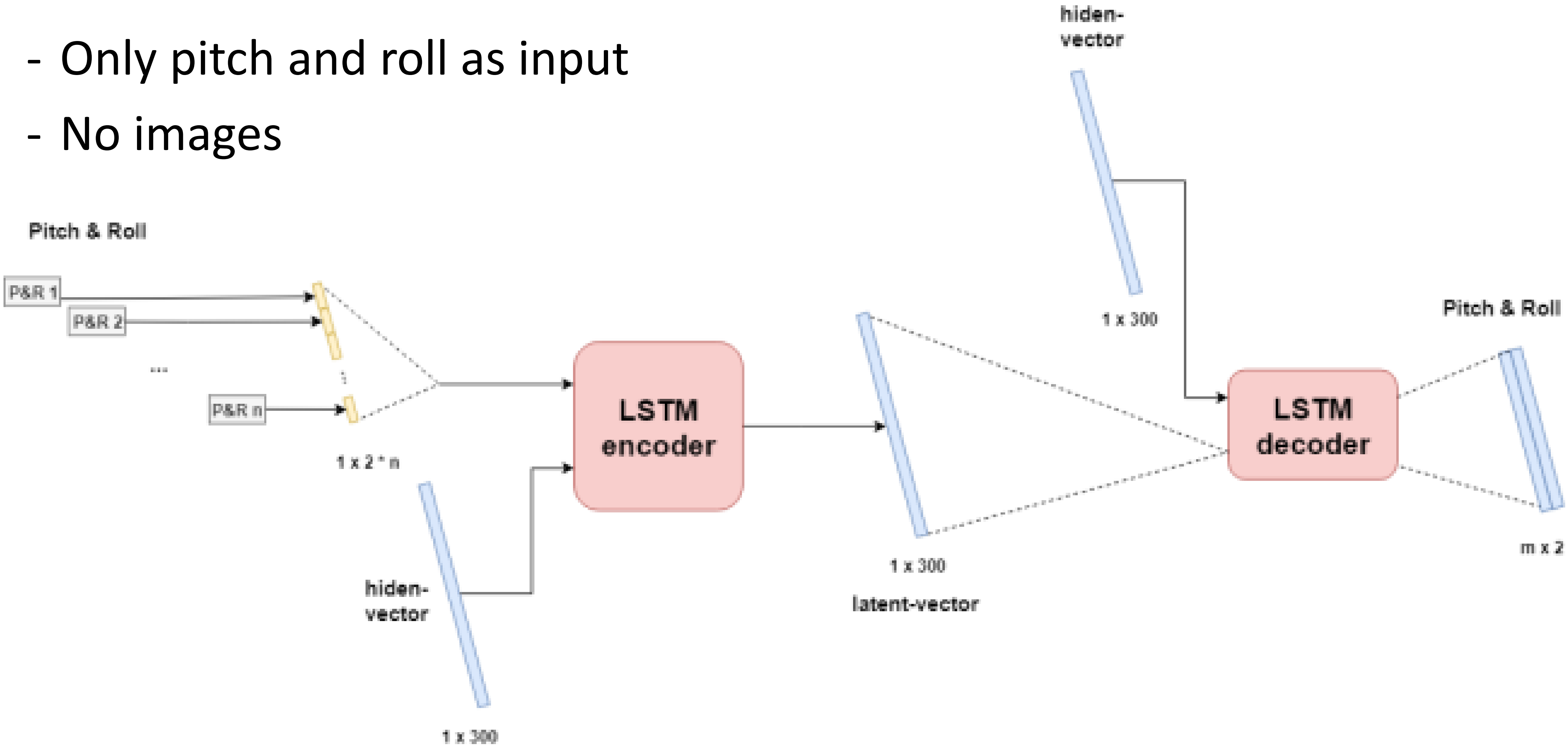


GHENT UNIVERSITY

Baseline model

LSTM encoder-decoder PR

- Only pitch and roll as input
- No images



LSTM encoder-decoder PR - performance

- Worst results
- Maximum pitch and roll in generated dataset: 62°

MODEL	MSE train x 10^2 avg over sum pitch and roll	MSE validate x 10^2 avg over sum pitch and roll	MSE test x 10^2 avg over sum pitch and roll
LSTM encoder decoder PR	0.855	0.869	0.723

MODEL	MSE pitch at 10th sec [denormalized]	MSE pitch at 10th sec [denormalized]
LSTM encoder decoder PR	30.99°	29.44°

GHENT UNIVERSITY

Source: https://github.com/Nazotron1923/Deep_learning_models_for_ship_motion_prediction_from_images/tree/master/Pre

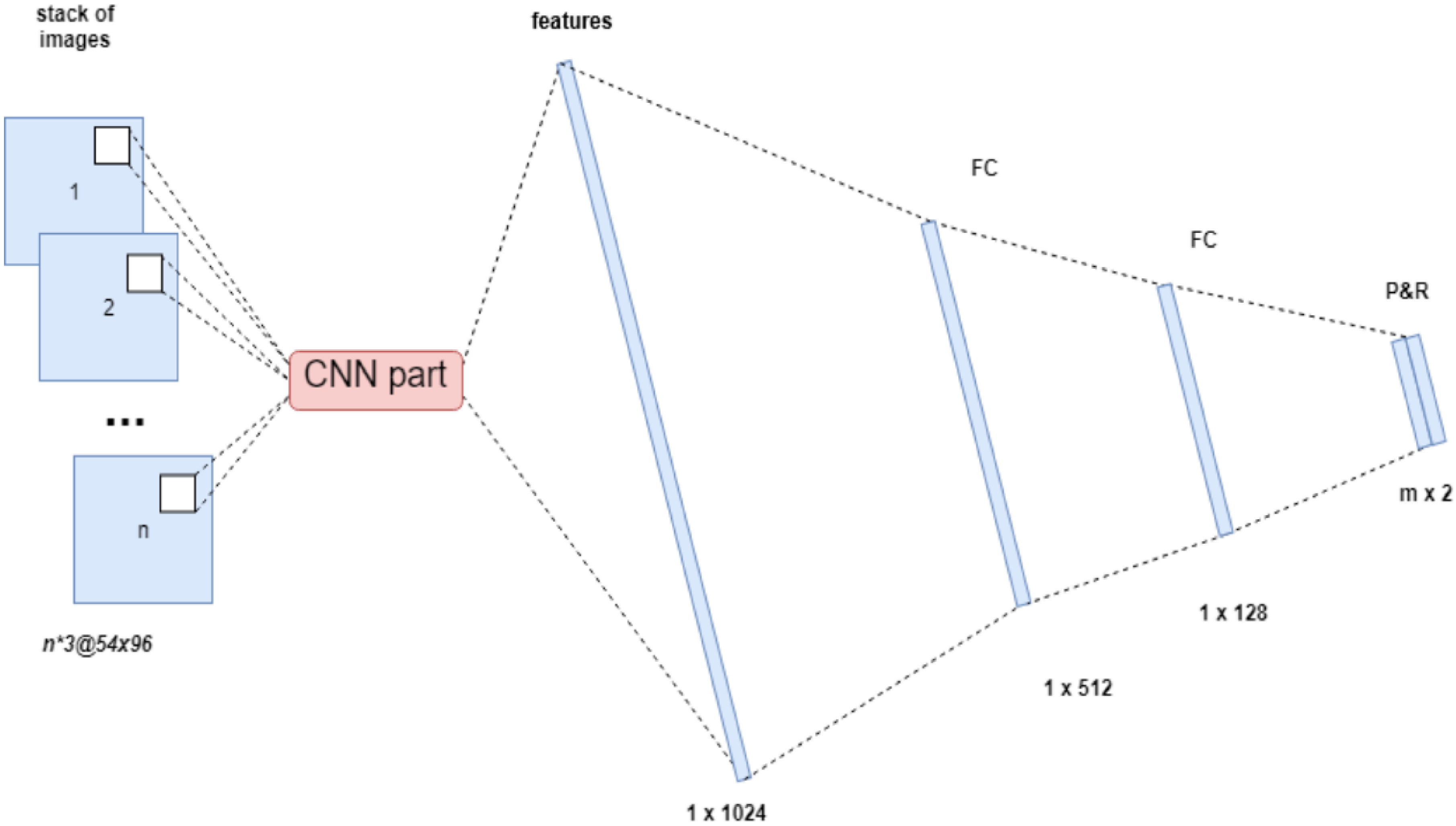
25

CNN Models

GHENT
UNIVERSITY

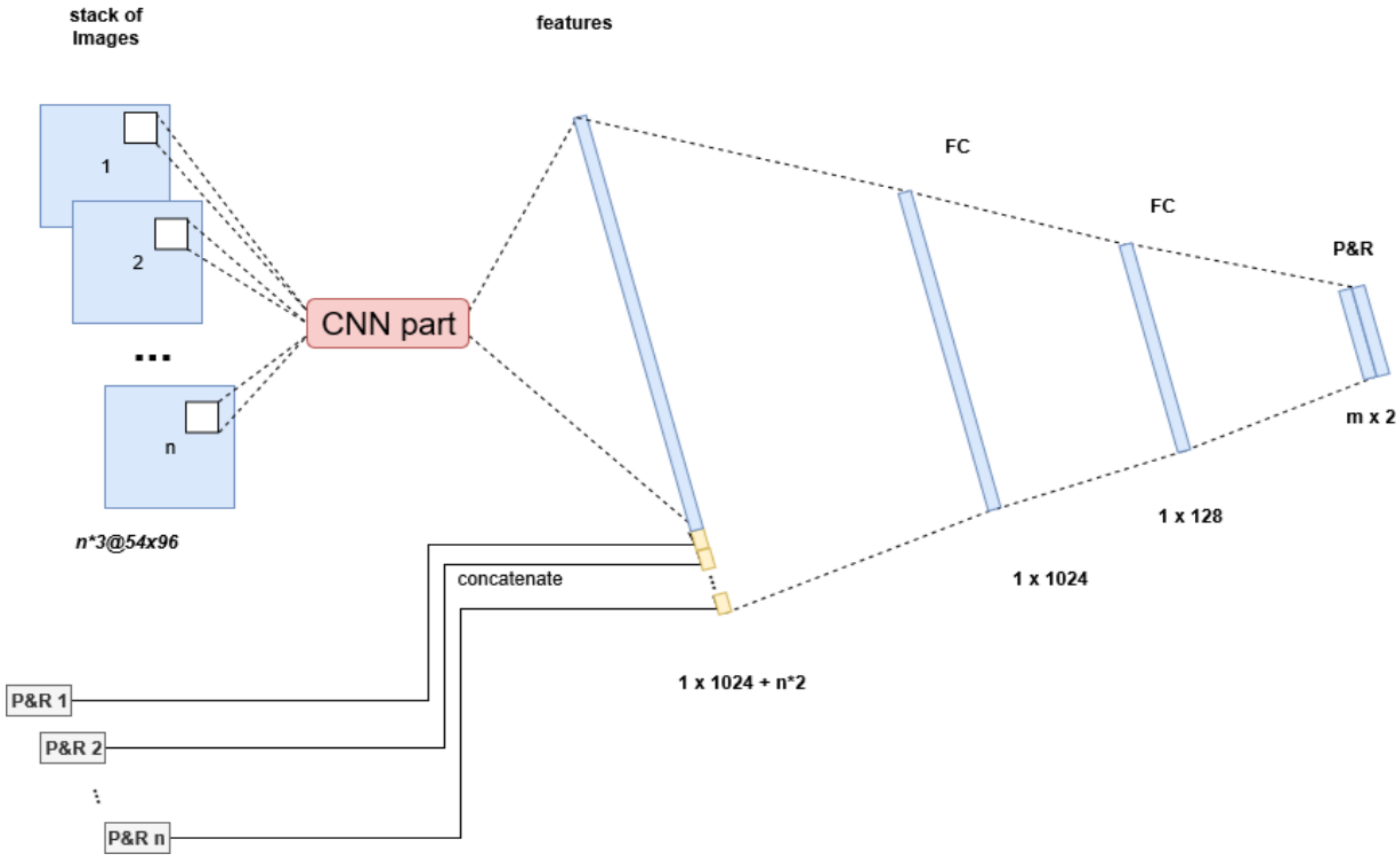
CNN Stack FC model

- Fully connected CNN
- Input
  - Stacked images (n)



CNN Stack PR FC model

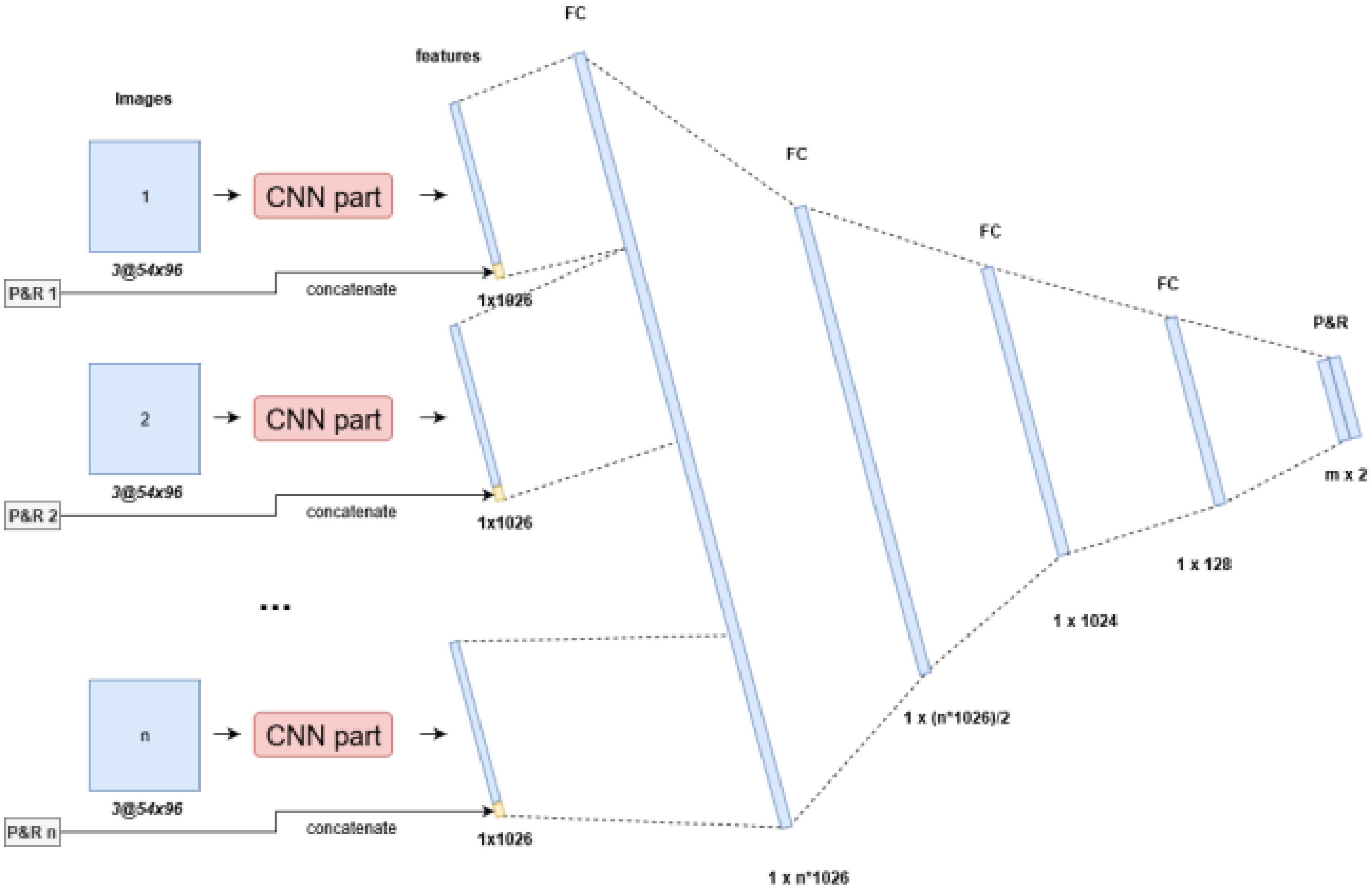
- Fully connected CNN
- Input
  - Stacked images (n)
  - **Pitch and roll**



GHENT UNIVERSITY

CNN PR FC model

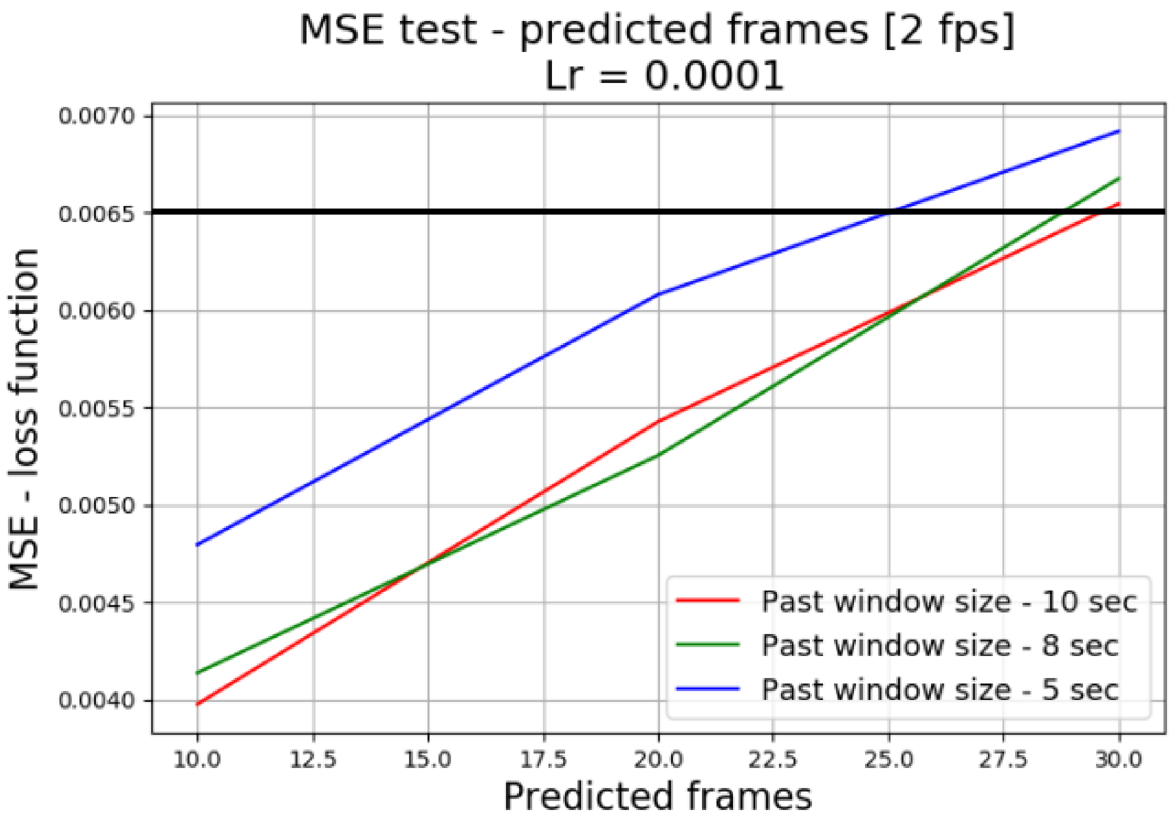
- Fully connected CNN
- Input
  - Pitch and roll
  - **Sep**. images (n)



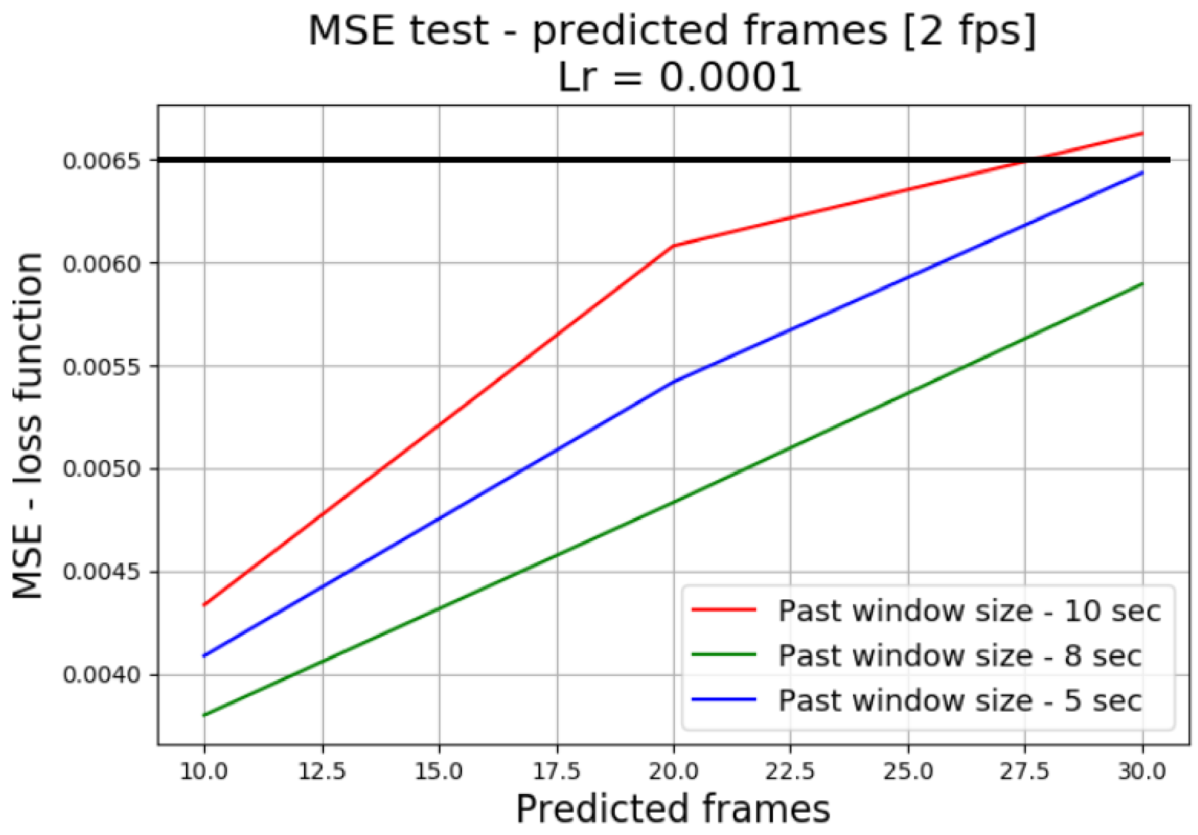
GHENT UNIVERSITY

CNN models - performance

- Without PR



- With PR input



CNN models - performance

- Separate images greatly improves accuracy

MODEL	MSE train x 10^2 avg over sum pitch and roll	MSE validate x 10^2 avg over sum pitch and roll	MSE test x 10^2 avg over sum pitch and roll
LSTM encoder decoder PR	0.855	0.869	0.723
CNN stack FC	0.635	0.647	0.619
CNN stack FC PR	0.556	0.470	0.483
CNN FC PR	0.378	0.259	0.252

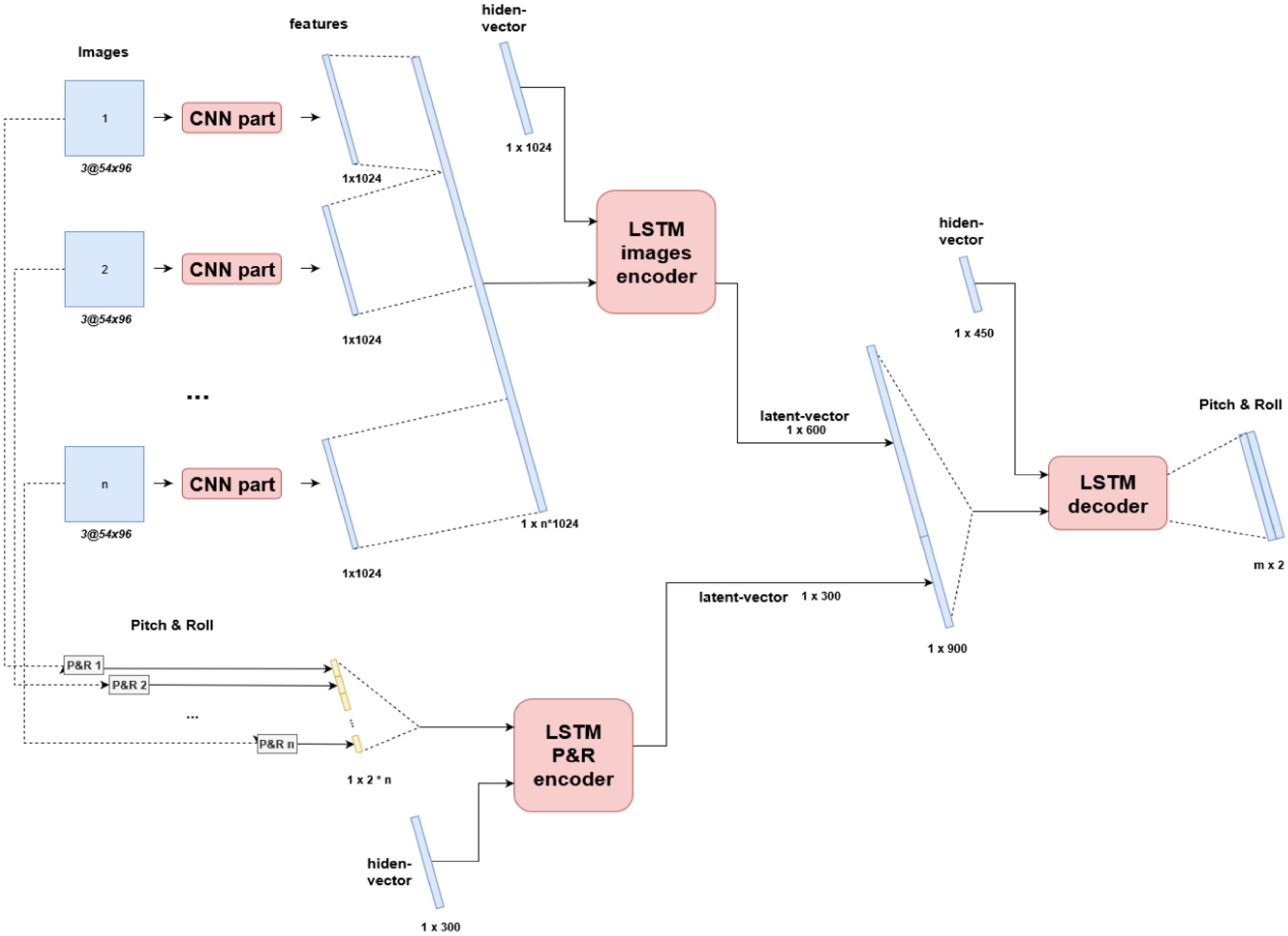
GHENT UNIVERSITY

LSTM Models

GHENT UNIVERSITY

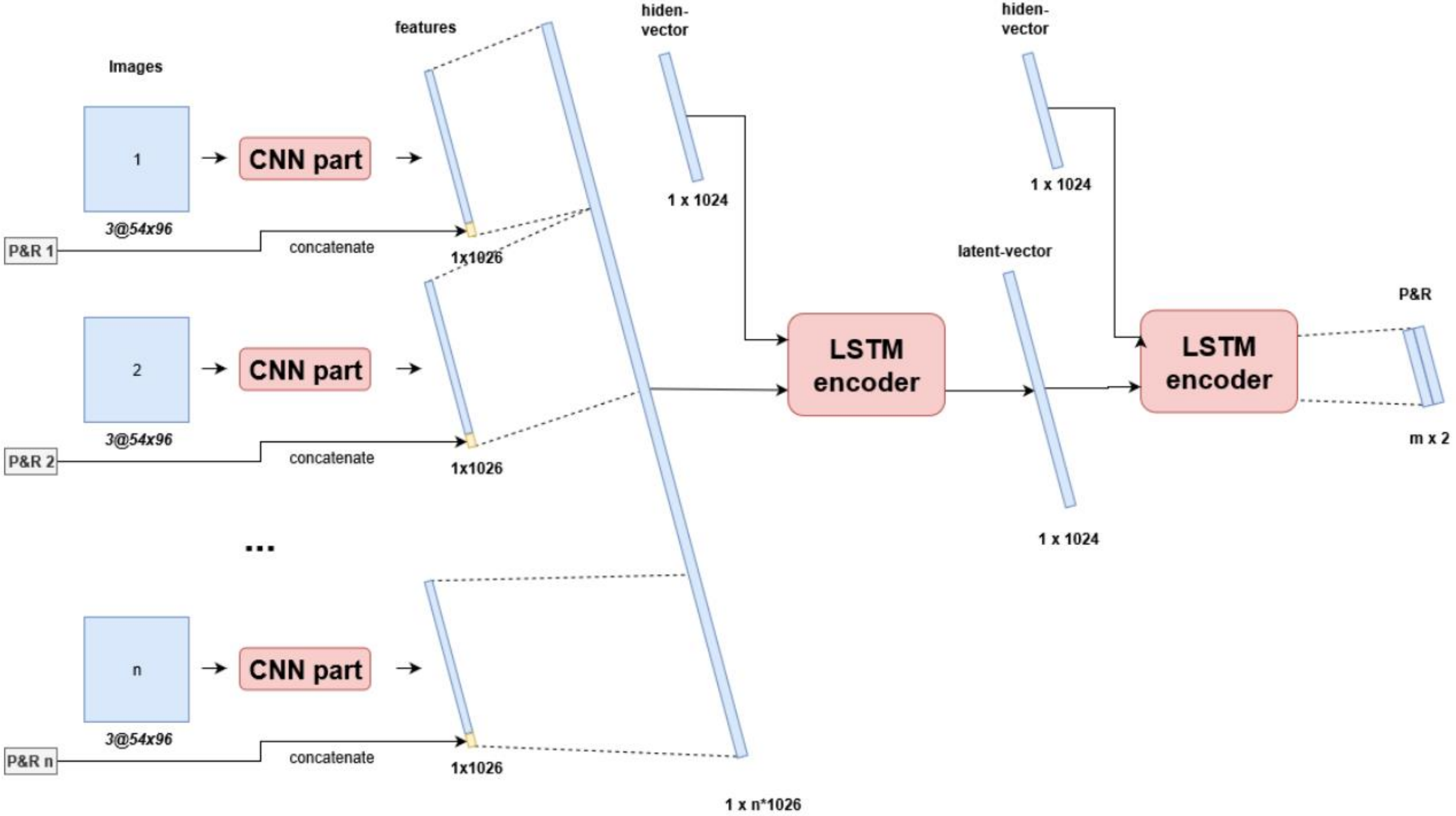
CNN LSTM img-encoder PR-encoder decoder

GHENT UNIVERSITY



GHENT UNIVERSITY

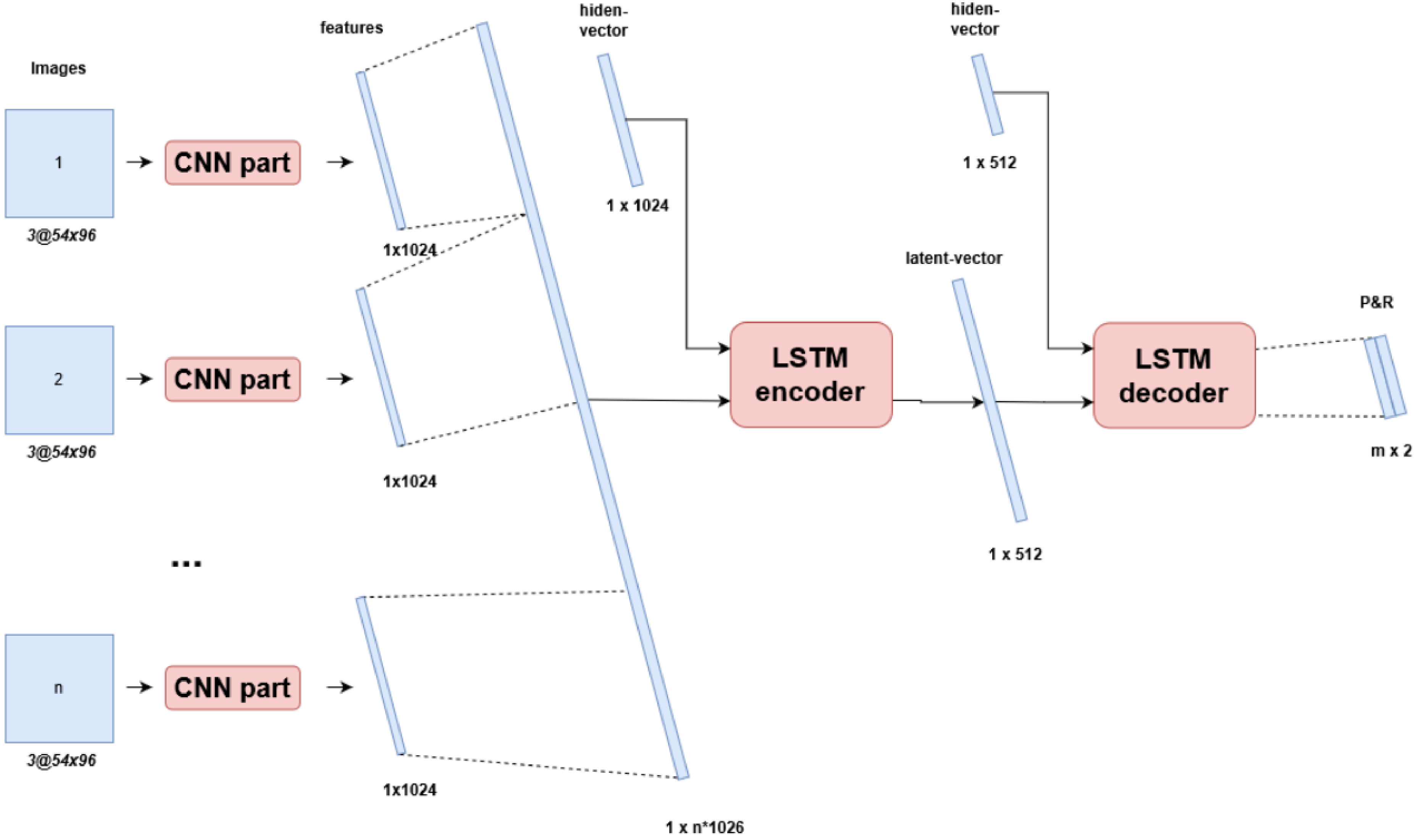
CNN LSTM encoder decoder images PR



GHENT UNIVERSITY

CNN LSTM encoder decoder image

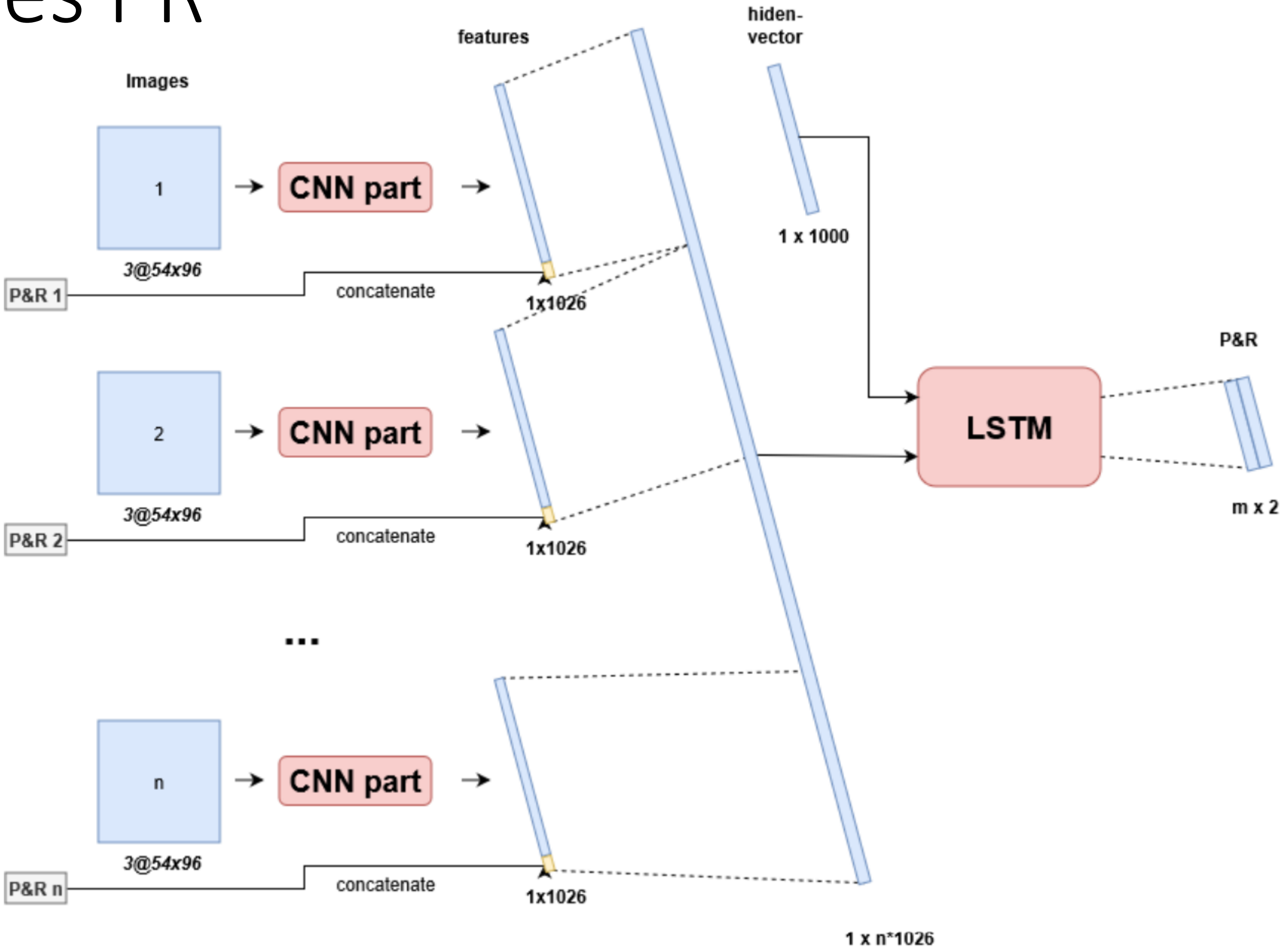
- PR dropped



GHENT UNIVERSITY

CNN LSTM images PR

- Reduce parameters



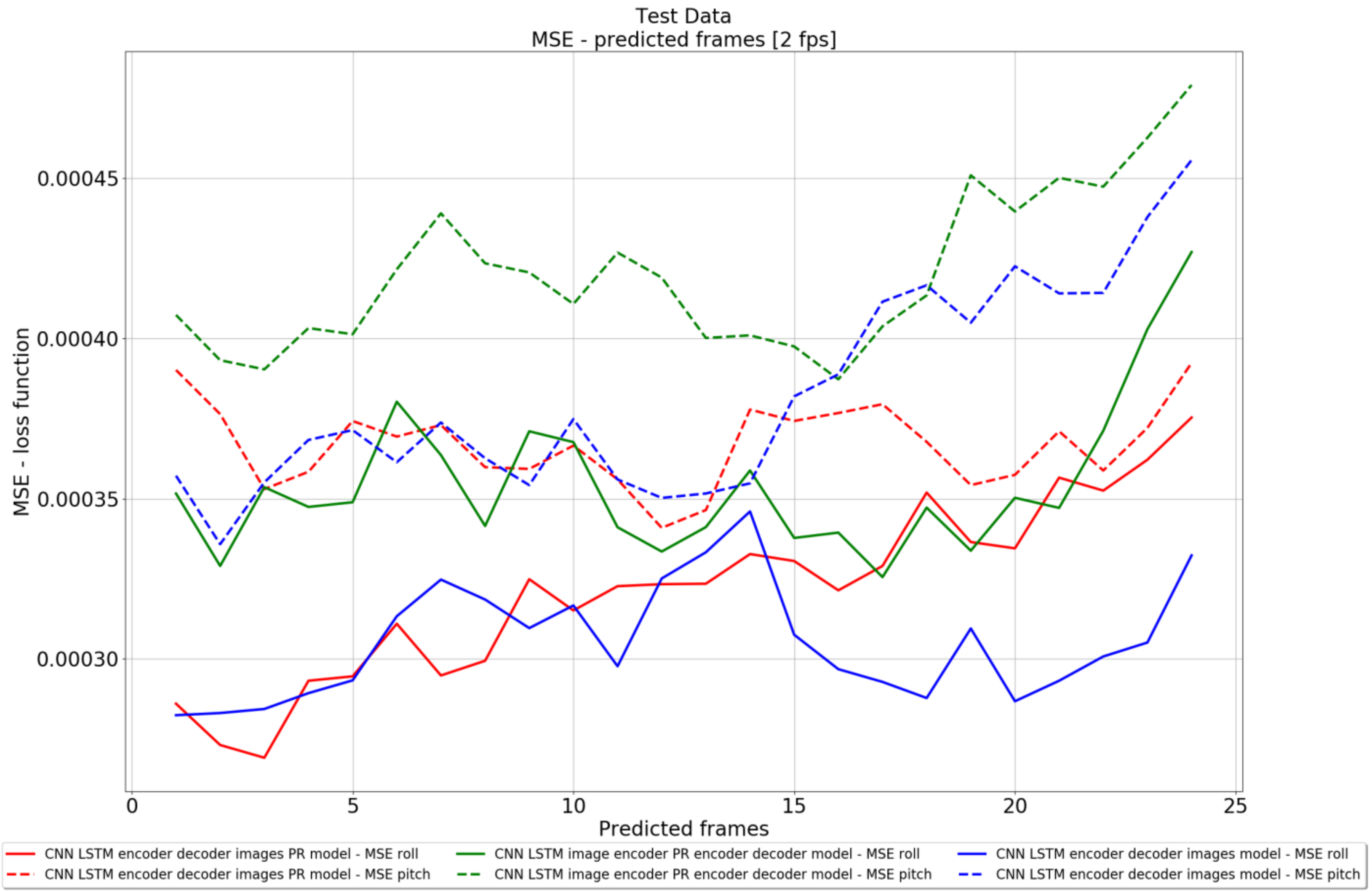
GHENT UNIVERSITY

LSTM models - performance

MODEL	MSE train x 10^2 avg over sum pitch and roll	MSE validate x 10^2 avg over sum pitch and roll	MSE test x 10^2 avg over sum pitch and roll
LSTM encoder decoder PR	0.855	0.869	0.723
CNN stack FC	0.635	0.647	0.619
CNN stack FC PR	0.556	0.470	0.483
CNN FC PR	0.378	0.259	0.252
CNN LSTM image-encoder PR-encoder decoder	0.044	0.104	0.080
CNN LSTM encoder decoder images PR	0.039	0.089	0.072
CNN LSTM encoder decoder images	0.038	0.091	0.071
CNN LSTM decoder images PR	0.126	0.138	0.108

Source: https://github.com/Nazotron1923/Deep_learning_models_for_ship_motion_prediction_from_images/tree/master/Pre

LSTM models - performance



GHENT UNIVERSITY

Denormalized results

Maximum values:

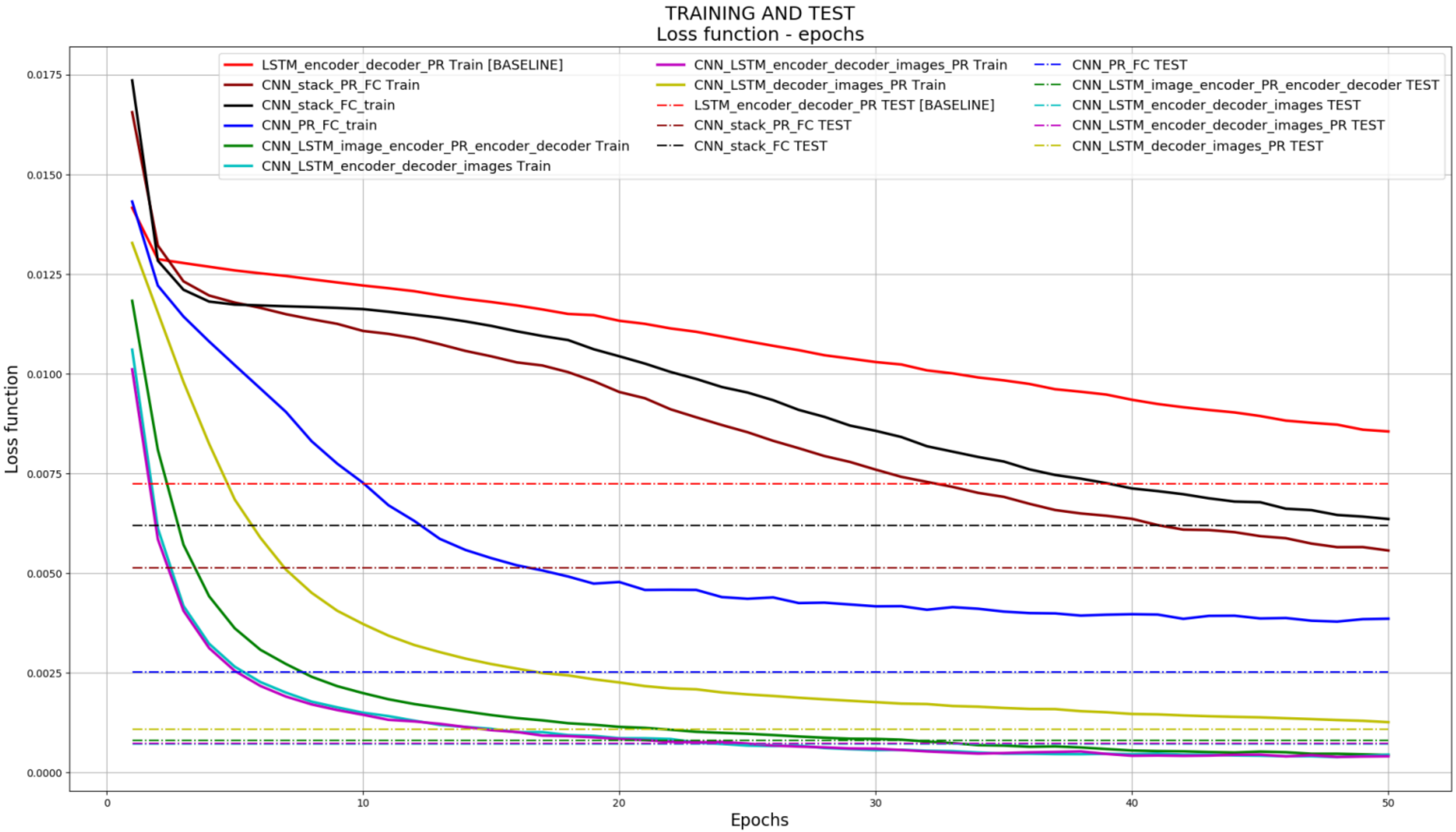
- Pitch: 61°
- Roll: 62°

- Baseline:
  - Avg: ~30°
  - 50% of max
- Reference:
  - Avg: ~3°
  - 5% of max

MODEL	MSE pitch at 10th sec [denormalized]	MSE pitch at 10th sec [denormalized]
LSTM encoder decoder PR	30.99°	29.44°
CNN stack FC	28.97°	28.94°
CNN stack FC PR	21.18°	22.56°
CNN FC PR	11.36°	11.28°
CNN LSTM image-encoder PR-encoder decoder	3.56°	2.83°
CNN LSTM encoder decoder images PR	2.89°	2.70°
CNN LSTM encoder decoder images	3.42°	2.32°
CNN LSTM decoder images PR	4.22°	4.16°

Source: https://github.com/Nazotron1923/Deep_learning_models_for_ship_motion_prediction_from_images/tree/master/Pre

Convergence



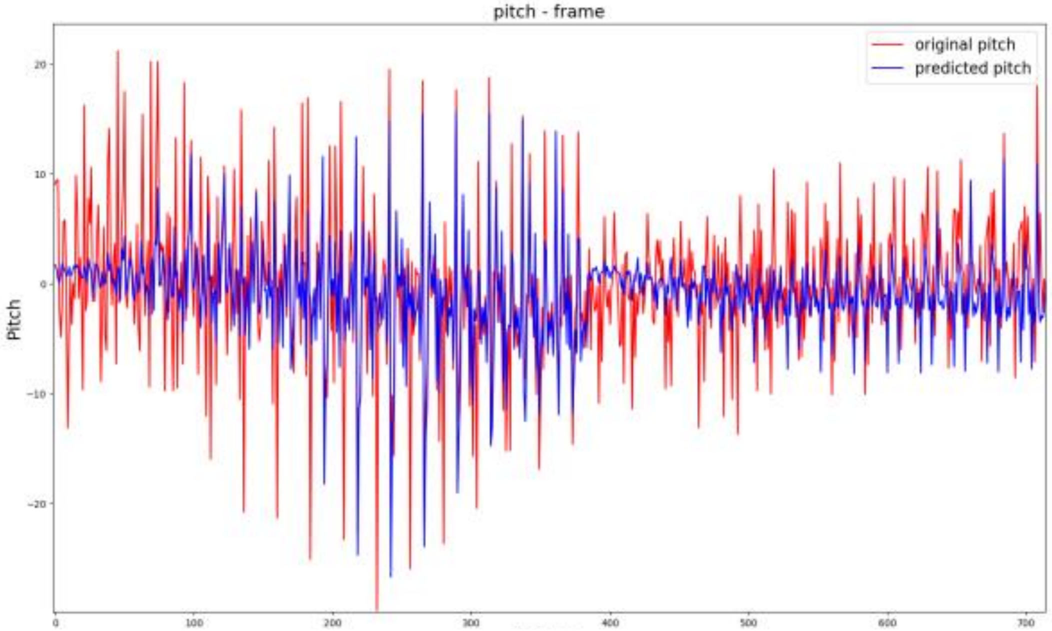
GHENT UNIVERSITY

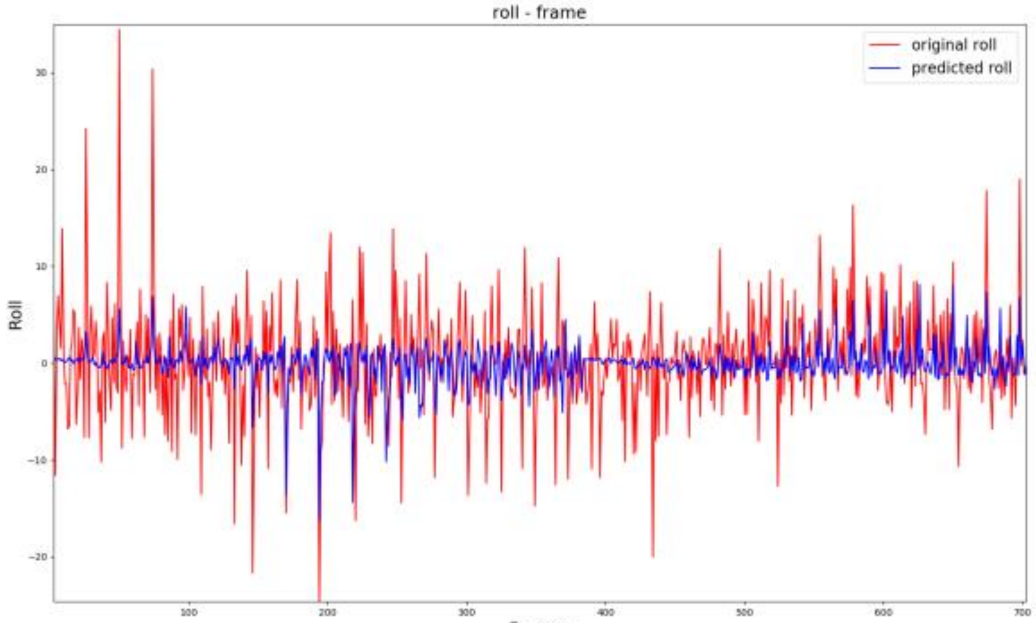
Best model

- Compare accuracy over large amount of predicted frames

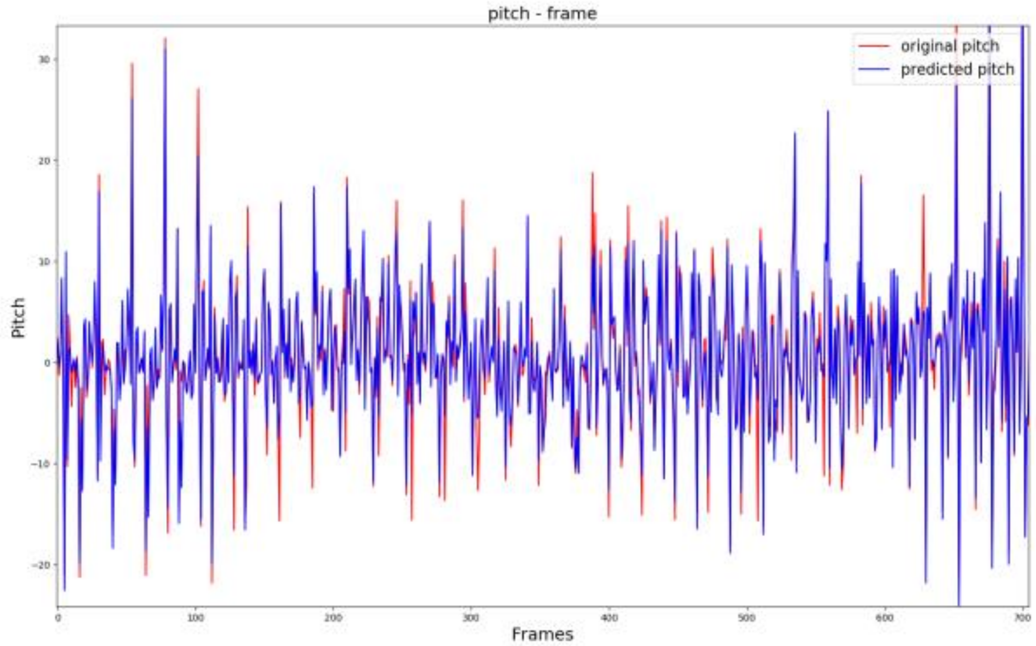
GHENT UNIVERSITY

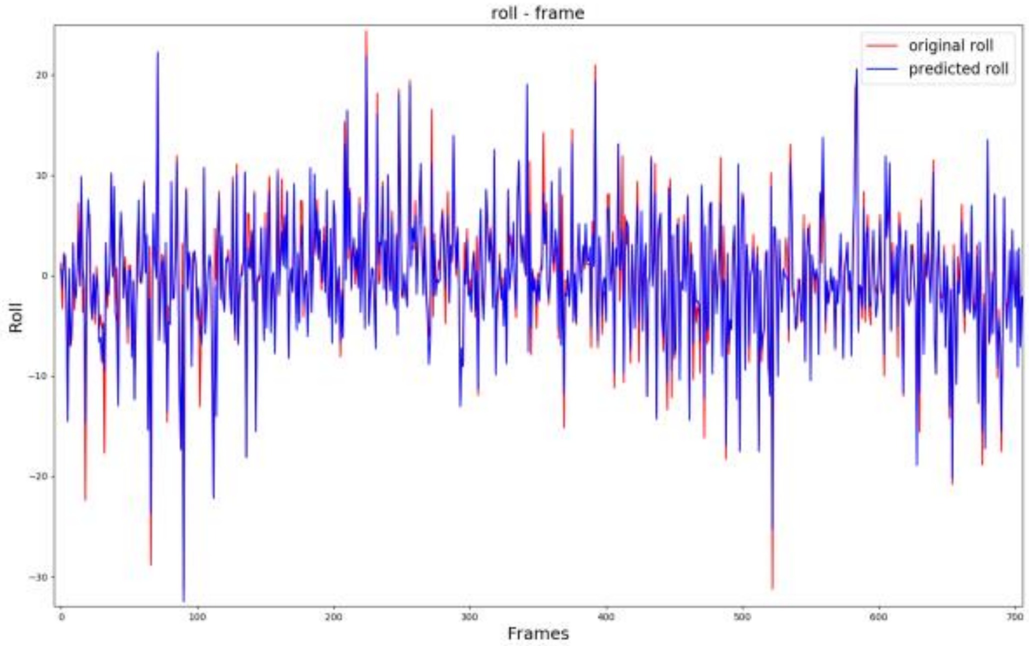
Without PR





With PR





GHENT UNIVERSITY

Hyperband optimization

- Test best model with optimized hyperparameters

Hyperparameters	**Best configuration**
Encoder latent vector size	700
Decoder latent vector size	408
Learning rate	$1.937 \cdot 10^{-4}$
Weight decay	$9.47 \cdot 10^{-5}$

- Final test up to 30 seconds

	At 15^{th} second	At 30^{th} second
Pitch MSE [denormalized]	2.9282012	3.7539778
Roll MSE [denormalized]	3.0667821	4.9226913

GHENT UNIVERSITY

44

Conclusion

- CNN LSTM encoder decoder images PR
- Criteria
  - MSE avg [0.0004; 0.0009] good results
  - 3° / 5% accuracy at 10th second
  - 4° / 7% accuracy at 15th second
  - 5° / 8% accuracy at 30th second

CNN LSTM encoder decoder images PR	0.00039 MSE	0.00089 MSE	0.00072 MSE
	10^{th} second	15^{th} second	30^{th} second
Pitch	2.89°	2.92°	3.06°
Roll	2.71°	3.75°	4.92°

GHENT UNIVERSITY

Implementation (2^{nd} semester)

- Test against 0 baseline
- Use existing models with augmented data
- Use existing models with real data
  - Define time criteria: real-time predictions
  - Extra output parameter: heave
- Compare results
- Develop and test new models
  - Compare with criteria
  - GRU + CNN
  - Transformer + CNN: GPT3

GHENT UNIVERSITY

Questions

GHENT
UNIVERSITY

Sources

- Deep Learning For Predicting Ship Motion From Images – Nazar-Mykola Kaminskyi - https://github.com/Nazotron1923/Deep_learning_models_for_ship_motion_prediction_from_images/
- Real-Time Ship Motion Forecasting Using Deep Learning – 2021 – Mohammad Hasanur Rashid, Jing Zhang, Minghao Zhao - https://dl.acm.org/doi/abs/10.1145/3448734.3450923
- A Neural Conversational Model – 2015 - Oriol Vinyals, Quoc Le - https://arxiv.org/abs/1506.05869
- A Co-operative Hybrid Model For Ship Motion Prediction - R. Skulstad, G. Li, T. I. Fossen, T. Wang, H. Zhang - https://www.researchgate.net/publication/351397594_A_Cooperative_Hybrid_Model_For_Ship_Motion_Prediction

GHENT UNIVERSITY