Deep Learning Model for Predicting Head Kinematics from Crash Videos

Vikas Hasija (✉ vikas.hasija@dot.gov)

Bowhead (Systems & Technology)

Erik G. Takhounts

National Highway Traffic Safety Administration (NHTSA)

Research Article

Keywords: Head kinematics, deep learning, brain injury risk

Posted Date: November 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1083278/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Head kinematics information is very valuable as it is used to measure brain injury risk. Currently, head kinematics are measured using wearable devices or instrumentation mounted on the head. These instrumentation and wearable devices can have errors due to faulty sensors and due to relative motion between the wearable device and the respective body region. This paper proposes a novel method to predict the head kinematics directly from videos without any instrumentation using a deep learning approach. To prove the concept, a deep learning model was developed for predicting time history of head angular velocities and their respective peaks using Finite Element (FE) based crash simulation data. This FE dataset was split into training, validation, and test datasets. A combined Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) based deep learning model was developed using the training and validations sets. The test (unseen) dataset was used to evaluate the predictive capability of the deep learning model. On the test dataset, correlation coefficient obtained between the actual and predicted peak angular velocities was 0.73, 0.85, and 0.92 for X, Y, and Z components respectively.

Introduction

In the United States, traumatic brain injury (TBI) is a serious public health issue. In 2014, about 2.87 million TBI related emergency department (ED) visits, hospitalizations and deaths occurred in United States [1]. Falls and motor vehicle crashes (MVC) were the first and second leading causes of TBI-related hospitalizations [1]. The lifetime economic cost of TBI, including direct and indirect medical costs, was estimated to be approximately \$76.5 billion (in 2010 dollars) [2]. Given the cost and number of TBI cases, understanding the mechanism of brain injury and preventing them is critical. Researchers over the years have found head motion kinematics to be an important correlate to brain injuries. Many head/brain injury metrics that have been developed such as the Head Injury Criterion (HIC) [3], Brain Injury Criterion (BrIC) [4], Rotational Injury Criterion (RIC) [5] etc. all make use of head motion kinematics. HIC is based on linear accelerations and is part of vehicle safety regulation [6], BrIC is based on angular velocities and RIC is based on angular accelerations. Measuring head kinematics is thus extremely important to understand the risk of brain injury.

The National Highway Traffic Safety Administration (NHTSA) commissions various types of research, New Car Assessment Program (NCAP) [7], and compliance crash tests which help determine the safety of a vehicle based on the injury metrics measured using anthropomorphic test devices (ATDs). The six degree of freedom (6DOF) kinematics of a 50th percentile male ATD head can be measured using different instrumentation, such as nine-accelerometer array sensor package (NAP) [8] or 6aw [9]. NAP makes use of nine accelerometers whereas 6aw makes use of six accelerometers and three angular rate sensors. While unlikely, the head kinematics obtained using these methods could have measurement errors due to faulty sensors, which can make it difficult to compute brain injury metrics and thus the risk of brain injury. NHTSA has conducted numerous Federal Motor Vehicle Safety Standard (FMVSS) No. 208 frontal crash tests [10] over the years. The 5th percentile female and 50th percentile male ATD used in these tests have limited head instrumentation to measure only linear accelerations at the head center

of gravity (CG). Also, smaller ATD headforms like the 3-year-old child ATD have limited instrumentation to measure only linear accelerations. Since angular kinematics are not directly measured in these ATD's, it is currently not possible to compute the angular head kinematics-based injury metrics such as BrIC and RIC.

In addition to crash tests with ATDs, to understand human biomechanical response during impact both human volunteer and post mortem human subjects (PMHS) tests are conducted by researchers. For the human volunteer tests, head kinematics are measured using wearable technology such as intelligent mouthguards [11] and for the PMHS tests, head kinematics are measured using devices such as pyramid nine accelerometer Package (PNAP) attached to the skull [12]. The problem with these devices is the relative motion between the sensors and the corresponding body part (skull, mouth or any other part), which leads to measurement errors in head kinematics. The higher the load, the higher the measurement error due to greater slippage, which makes these measurements unreliable. Such instrumentation also requires a lot of time and effort to analyze the errors. Moreover, sensors are expensive and time consuming to attach and calibrate before every test.

Deep learning is a part of machine learning based on artificial neural networks and has been shown to be very effective in solving complex problems in the area of computer vision, natural language processing, drug discovery, medical image analysis, etc. Recently, deep learning models were used in biomechanics field as well. Wu et al. [13] used American college football, boxing and mixed martial arts (MMA) datasets along with lab reconstructed National Football League impacts with FE head Injury model to develop a deep learning model to predict 95th percentile max principal strain of the entire brain and the corpus callosum along with fiber strain of the corpus callosum. Zhan et al. [14] used kinematic data generated by FE simulations and those collected from on-field football and MMA using mouth guards and developed a deep learning head model to predict the peak max principal strain of every element in the brain.

Although deep learning models have been used to predict brain deformations from head kinematics, no study has focused on using deep learning to predict head kinematics. This paper proposes a novel deep learning approach to predict head kinematics directly from crash videos. As a proof of concept, this study developed a deep learning model to predict X, Y and Z-components of head angular velocity vector from FE based crash videos. These predicted angular velocities were used to compute the head angular kinematics-based injury metric, BrIC, for comparing actual and predicted BrIC values.

Methods

Data: A supervised deep learning model takes in the inputs and the corresponding outputs and learns the mapping between the inputs and the outputs. For developing a deep learning model for predicting time history of angular velocities from crash videos, crash videos are required as inputs and the corresponding time history of angular velocities are required as outputs. There are no such real-world datasets available. Data can be taken from the NHTSA commissioned crash tests where crash test videos can be used as inputs and the corresponding head angular velocities measured on the ATD can be used as outputs to

develop a deep learning model. However, there is very limited data from limited test conditions. Hence, FE based crash simulation data was utilized in this proof of concept study.

To generate the data, validated simplified Global Human Body Models Consortium (GHBMC) 50th percentile male [15, 16] and 5th percentile female [17, 18] FE human models were used. These human models were positioned in the driver compartment (Figure 1) that was extracted from the validated FE model of a 2014 Honda Accord [19].

A validated generic seatbelt system with retractor, pretensioner and load limiter was included in the model along with validated frontal and side airbags [19]. In addition, steering column collapse was implemented and was included in these simulations. The roof rails, side door, B-pillar, and floor were deformable in the full FE model, but were made rigid in this study. The knee bolster and A-pillar were kept deformable. The human models were positioned in the driver compartment based on clearance measurements taken from physical crash tests (NHTSA test number 8035 for 50th male, NHTSA test number 8380 for 5th female; <https://www-nrd.nhtsa.dot.gov/database/veh/veh.htm>). The crash pulse used for the simulations was taken from the physical crash test and is shown in Figure 2.

These human models were evaluated in full frontal test condition, following which a design of experiments (DOE) study was conducted. For the DOE study, both crash-related parameters and restraint-related parameters were varied (Table 1). The crash related parameters were Delta-V and principal direction of force (PDOF). The restraint parameters were both seatbelt and airbag related. The parameters were varied over a wide range to generate a range of head motions including cases where the head hits the steering wheel.

Parameter	Range
<i>Crash-related Parameters</i>	
Delta-V	25 mph - 45 mph
PDOF	-30° (near side) – 30° (far side)
<i>Restraint-related Parameters</i>	
Frontal & side airbag mass flow rate	± 25 %
Frontal & side airbag firing time	5 ms- 70 ms
Collapsible column breaking force	3000 N- 10000 N
Load limiter	1000 N- 5000 N
Pretensioner limiting force	1000 N- 3000 N
Friction between head and front/side airbag	0 - 3

Table 1. Parameters and their ranges

The crash pulse for the same vehicle may be different for different PDOF, frontal overlap, and type and stiffness of the impacting surface. In addition, for the same PDOF, frontal overlap, and impacting surface, the crash pulse can vary for different vehicles of the same size (e.g., mid-size sedans). To keep the number of variables manageable for the DOE study, crash pulse shape was kept constant. Only crash pulse magnitude was scaled to achieve different Delta-Vs.

A total of 1010 scenarios were simulated covering a wide range of crash conditions. Each crash scenario was simulated for a duration of 150 ms. For each simulation, the time history of head angular velocities was computed and four crash videos with different views were generated (Figure 3).

The views chosen were similar to camera views available from NHTSA crash tests. Since the aim of the study was to predict the time history of head angular velocities from any view, each crash view was treated as a separate sample. Thus, we had a total of 4040 crash videos and their corresponding head angular velocity time histories (ω_x , ω_y , ω_z) for each of the three head rotational axes. The crash videos were then used as inputs for the deep learning model and the corresponding angular velocity time histories were used as the “ground truth” outputs. For the purposes of this study, all crash videos were generated such that only the human model was visible. The vehicle structure and the airbags were removed from the videos to prevent any head occlusion.

Since videos are used as inputs to the deep learning model in the form of sequence of images, an additional input pre-processing step was carried out to convert the FE based crash videos to sequences of images. Given the goal of this study was to predict the time histories of head angular velocities, the motion of the head was extracted as a sequence of images over time from each FE crash video (Figure 4). These sequences of images were then used as inputs to the deep learning model.

To extract the motion of the head over time as a sequence of images, the head needs to be detected in each frame of the crash video. A head detection algorithm may be employed for this purpose. The use of FE based crash videos in our study offered an additional advantage in utilization of a fast and accurate computer vision-based color mask as a head detection algorithm. In all crash videos, the head of the human model was colored green and the rest of the body was kept gray such that the head could be easily detected in each frame with a bounding box using Contours in OpenCV [20]. Once detected, the head image inside the bounding box was extracted from each frame to obtain a sequence of head images over time. The images were extracted every 2 ms from the 150 ms crash event and thus each sequence of images had a length of 76. The corresponding “ground truth” time histories of angular velocities (outputs or targets) were also sampled every 2 ms to match the corresponding sequence of images. An example of the input and corresponding output for training the deep learning model is shown in Figure 5.

The Contours based detection technique gave zero false positives and generated a complete sequence without missing any frames. However, it only works if the user has a full control over all color aspects of

the videos, which was the case in our study. For a “real world condition” (not based on the simulations), a head detection model with a head tracking algorithm, such as Kalman filter, may be more appropriate.

Input data Transformation: The input data (sequence of images) were RGB (red, green, blue) images with a data type of “uint8.” A “uint8” data type contains pixel values in the range from 0 to 255 (pixel value of 0 corresponds to the darkest color in the range while value of 255 corresponds to the lightest value). Deep learning models train better and faster when input data is on the same scale. Thus, all the input sequences of images were normalized so that the pixel values were in the range from 0 to 1 with the data type of “float32.” Due to resource limitations, all images were resized to a height and width of 64 pixels and subsequently converted to grayscale such that each sequence of images had a shape of (76, 64, 64, 1), where number 76 stands for the number of images in a sequence, numbers 64 are for image size, and 1 represents a grayscale image (for color images the number 1 would’ve been replaced with number 3 for each channel of the RGB image).

Data splitting: The entire dataset had 4040 samples. The count plot in Figure 6a shows the distribution of data for each human model size and for each view. For developing the deep learning model, this dataset was split into three datasets: training, validation and test datasets. 74% the data was used for training, 13% of the data was used for validation and 13% of the data was used for testing. Data splitting was carried out using stratified sampling based on human model size and the crash view to ensure each of these (human model size and crash view) were equally represented in all three datasets (Figure 6b).

The training and validation datasets (87% of the data) were used for model development. The validation dataset was used for hyperparameters tuning and was a part of model development. The test dataset was not used in model development and was treated as an unseen dataset that was used to evaluate the final performance of the model.

Deep learning model: The overall architecture for a deep learning model depends on the type of input data. The input data in this study is a sequence of images over time. Convolutional Neural Networks (CNN) can capture spatial dependency and are one of the most common types of neural networks used in [computer vision](#) to recognize objects and patterns in images. On the other hand, Recurrent Neural Networks (RNN) can capture temporal dependency and are commonly used for sequential data processing. Thus, to process sequences of images in this study, a deep learning model that combines CNN [21] and Long Short-Term Memory (LSTM) based RNN [22] was used (Figure 7). The CNN-LSTM architecture uses CNN layers for feature extraction on input data combined with LSTMs to support sequence prediction.

Since the best architecture for our problem was not known at the start of model development, a lightweight baseline model (with fewer trainable parameters) was developed, which was later improved using hyperparameter tuning. For the CNN part of the baseline model, a Visual Geometry Group (VGG) style architecture [23] was used, which consisted of a three-block network with two convolutional layers per block followed by a max pooling layer. Batch Normalization [24] and a Rectifier Linear Unit (ReLU) activation function [25] were used after each convolutional layer. The baseline (initial) values selected for

the number of convolutional filters for the three blocks were 16, 32 and 64 respectively. A global average pooling layer was added as the last layer of the CNN model to obtain the feature vector. Since each input sample is a sequence of images, the CNN part of the model was wrapped in a Time Distributed layer [26] to get feature vector corresponding to the entire sequence. The Time Distributed wrapper helps apply the same CNN network to every temporal slice (image) of the input. The output from CNN was used as an input to the LSTM network.

For the LSTM part of the baseline model, one LSTM layer with a hidden size of 128 was used. Since input sequence has a length of 76 and the goal is to predict the time history of angular velocity, the output was obtained at each recurring timestep from the LSTM layer. The output of the LSTM was then used as an input for a fully-connected layer with the ReLU activation function, followed by a Dropout layer [27] to control for overfitting. The output of the dropout layer was then fed to a fully-connected layer with a linear activation function to generate the final output, i.e. the predicted time history of angular velocity. Linear activation generates continuous numerical values and hence was used in the final output layer as angular velocity time history prediction was solved as a regression task.

The Mean Squared Error (MSE) between the actual and predicted time history was used as the loss function for training the entire model. Adaptive moment estimation (Adam) optimizer [28] was utilized for optimization. Since the ReLU activation was used in the network, He-Normal initializer [29] was used to initialize the trainable weights of the model. The model was developed using Tensorflow v2.4 [26].

Individual deep learning models and Training: Since the three components of angular velocity (ω_x , ω_y , ω_z) are independent of each other, three separate deep learning models were trained one for each component of angular velocity ω_x , ω_y , and ω_z (as opposed to training one deep learning model for predicting time history of all three components of angular velocity. The same training and validation inputs were used for training all three models. Only the “ground truth” targets were changed depending on the model. The baseline models for ω_x , ω_y , and ω_z were trained with a learning rate of 0.0001 and with a batch size of 8 for a maximum of 80 epochs. Early stopping [26] with a patience of 10 and model checkpointing [26] callbacks were used to save the best model based on validation loss. Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. For this purpose, ReduceLROnPlateau callback [26] was utilized. This callback monitors the validation loss and if no improvement is seen for 5 epochs, the learning rate is reduced.

The hyperparameter values chosen for the CNN, LSTM, and the extended part of the baseline model were selected at random and did not necessarily correspond to the best architecture for the problem. To improve the models, hyperparameter tuning was carried out to find the set of hyperparameter values that give the best results for our problem. Validation loss was tracked to find the best set of hyperparameters. Table 2 shows the hyperparameters that were varied along with their corresponding range.

Hyperparameters	Baseline value	Range explored
CNN Based		
Number of VGG blocks	3	1 – 5
Number of convolutional filters per block	16,32,64	16 - 64
Pooling type	Max	Max, Average
LSTM Based		
Number of LSTM layers	1	1-2
Number of LSTM units	128	64 - 256
Others		
Number of units for fully-connected layer	80	64 - 128
Dropout rate	0.5	0.0 – 0.5
Learning rate	1e-4	1e-4 – 1e-2

Table 2. Hyperparameters

Keras-Tuner [30] was used for hyperparameter tuning using Bayesian Optimization [31]. Because of resource limitations, hyperparameter tuning was only performed for the ω_x model to find the best set of hyperparameters. This set of hyperparameters was then used to train the final deep learning models for all three components of angular velocity.

Combined Model: The three individually trained models for ω_x , ω_y , and ω_z were combined into a single deep learning model as shown in Figure 8. To predict the time history of the three components of angular velocity from a video input of any view, the video (preprocessed as sequence of images) is passed into the combined model. It is then propagated (forward pass) through the individually trained networks that output time history of the three components of angular velocity ω_x , ω_y , and ω_z .

Model evaluation: The three individually trained deep learning models for ω_x , ω_y , and ω_z were evaluated on the test dataset to see how well they generalize on unseen data. The actual and predicted time histories for cases from the test dataset were compared quantitatively using CORA [32]. While time histories of angular velocities are important to assess overall head kinematics, for computing brain injury metrics peak values are usually used. For example, Brain Injury Criteria (BrIC) [4] is computed using absolute peaks of ω_x , ω_y , and ω_z (equation (1)).

$$\text{BrIC} = \sqrt{\left(\frac{\omega_{xm}}{66.25}\right)^2 + \left(\frac{\omega_{ym}}{56.45}\right)^2 + \left(\frac{\omega_{zm}}{42.87}\right)^2} \quad (1)$$

To evaluate prediction of the peak angular velocity, correlation coefficient between the actual and predicted peaks was computed for all three models using the test dataset.

The combined model was also evaluated for a few cases from the test dataset. For the combined model, in addition to comparing the time histories using CORA, the actual and predicted BrIC values were also compared.

Results

Individual model evaluation

The final deep learning models (with tuned hyperparameters) for ω_x , ω_y , and ω_z were assessed on the test dataset to evaluate how well they generalize on unseen data. Figure 9 shows the actual and predicted time histories for ω_x for 5 randomly selected cases from the test dataset along with their respective CORA score. It can be observed that the ω_x deep learning model is able to predict the time histories for most cases. Regarding the peak values, for some cases predictions were reasonable (Figure 9a, 9e), while for others there was some mismatch (Figure 9b, 9c, 9d).

Figure 10 shows the actual and predicted time histories for ω_y for 5 randomly selected cases from the test dataset along with their respective CORA score, which demonstrates better prediction of the time histories and the peaks for ω_y component than those for ω_x .

Similarly, Figure 11 shows the actual and predicted time histories for ω_z for 5 randomly selected cases from the test dataset along with their respective CORA scores demonstrating a reasonable match between actual and predicted peaks and time history of ω_z . Additionally, Figure 11 shows different types of head impacts: 1) head contacts the airbag (Figures 11a, 11b, 11d, 11e), and 2) head contacts the steering wheel (Figure 11c). The model is able to predict the peaks and time history for these different head impacts demonstrating that the trained deep learning models are capable of learning important features that distinguish between different types of head impacts.

The peak angular velocities were evaluated quantitatively (Figure 12) with correlation plots using only the unseen test dataset. Correlation coefficients were 0.73 for ω_x , 0.85 for ω_y and 0.92 for ω_z .

Combined model evaluation

The combined model performance was also assessed on the test dataset. Three random videos with different views (back view, front view and left side view) were selected from the test dataset and the 3D angular kinematics predicted by the combined model were compared with the actual data. The actual and predicted time history results for these three videos along with their respective CORA score are shown in Table 3. The actual and predicted BrIC values are also included in Table 3.

Discussion

The objective of this study was to develop a novel approach using deep learning to predict head angular kinematics directly from crash videos without the use of sensors. Deep learning models are function approximators that approximate the unknown underlying mapping function from inputs to outputs. The results of this study show that deep learning models for ω_x , ω_y , and ω_z are capable of effectively capturing important features from the input sequence of images and mapping them to the output angular velocities.

Deep learning models have been shown to work well on unstructured data, such as images, videos, text etc. Depending on the task, deep learning models can have a large number of trainable parameters (tens of millions) and are thus trained on large datasets to avoid overfitting. For example, image classification models are trained using the popular ImageNet dataset that has millions of images. In this study, the data size of 4040 was generated using FE simulations. Since dataset size is small, in addition to using regularization techniques like dropout to avoid overfitting, the number of trainable parameters were also kept under check. The final deep learning model had approximately ~850K trainable parameters. Despite using this rather small dataset, the deep learning model effectively captured the mapping between input and output. Similar sized biomechanical datasets have been used in the past for training deep learning models. Wu et al [13] used a dataset of size 3069 to develop a deep learning model for efficient estimation of regional brain strains and Zhan et al [14] used data from 1803 head impacts for developing their deep learning model for estimation of entire brain deformation in concussion. Both these studies also relied on FE head models for mapping head kinematics to brain strains. Wu et al [13] study used Worcester Head Injury Model [33] whereas Zhan et al [14] used KTH model [34] to calculate brain MPS based on input head kinematics.

In this study, the data for developing the models was generated using a range of test and restraint conditions (Table 1). This was necessary to yield a wide range of head motions. The data was not uniformly distributed for each condition. Despite this the deep learning models for ω_x , ω_y , and ω_z were capable of predicting the time histories and their respective peaks (Figures 9, 10, 11, 12). With addition of more training data for the underrepresented test conditions the model predictions may further improve.

The x-component of the angular velocity vector (ω_x) had more discrepancy in predicting peak values when compared to ω_y and ω_z (Figure 12). The reason for this discrepancy may be due to the views (left, left isometric, front isometric and back isometric) selected for training the models. These views may be more conducive for learning important features related to prediction of ω_y and ω_z , but not of ω_x . More analysis is necessary in finding specific camera angles (views) that would work better for all components of angular velocity.

Due to resource limitations, low resolution (64,64) grayscale images were used, and limited hyperparameter tuning was carried out for the ω_x model. These same hyperparameters were then used for final training of the ω_y and ω_z models. The ω_x model may be further improved by exploring a wider range of hyperparameters, while the ω_y and ω_z models may be improved by conducting hyperparameters tuning specifically for each of these models. In addition, using more advanced CNN architectures like

Residual networks [35], EfficientNets [36] etc. instead of VGG together with higher resolution RGB images may help the networks learn better features from the input sequence of images, thus further improving the angular velocity predictions (time history and peaks).

Deep learning models work well with the type of data they are trained on (i.e. deep learning models may not be well suited for extrapolation). The models in this study were trained on a limited set of views (left, left isometric, front isometric and back isometric), hence they offer reasonable predictions only when the images are used from these specific camera angles.

The deep learning models in this study were trained using head images of the GHBMFC FE human model. If these trained models were fed input data from head images that are different (human or ATD head images) from what they are trained on, the models may not predict well due to changes in the input data domain. To use this approach for ATD related crash videos, either new deep learning models can be developed using ATD based data using the procedure described in this study or FE data based deep learning models can be modified using domain adaptation [37]. Domain adaptation uses data in source domain (FE data in our case) to solve tasks in a target domain e.g. ATD head images.

The deep learning models in this study were developed from FE crash videos with an unobstructed view of the head. Any head occlusion can lead to head features not being identified correctly, which can give inaccurate prediction of angular velocities. For real world crash tests, the head may get obstructed at some point during the crash event by the vehicle structure, airbags, etc., and for this methodology to work training of the deep learning models should be performed using data from camera locations/views in which the head remains visible (or partially visible) throughout the crash event, such as the back-isometric view in NHTSA crash tests. A deep learning model developed using such views can then be used to predict 3D rotational kinematics from similar 2D views as shown by the combined model results (Table 3).

Using this deep learning approach, 3D rotational kinematics can be obtained from the crash test videos involving 5th percentile ATD and other smaller ATD's like the 3-year old child ATD which have limited head instrumentation and currently don't have sensors to output rotational head kinematics. This deep learning approach only requires videos to make predictions and can be used to predict head kinematics for any size ATD. Similarly, this approach can be extended for predicting head kinematics from volunteer and PMHS tests videos.

Using sensors to obtain head kinematics from ATDs, PMHS and/or volunteer tests, is a time-consuming process as it requires sensor installation and calibration before each test. Training deep learning models, on the other hand, require high end machines with GPUs. However, once trained such models can be used to make angular velocity predictions in a few seconds from any given crash video. The advantage of deep learning approach is that with the availability of new data, the training dataset may be appended for retraining, thus allowing for iterative improvement of the model.

Biomechanical data is limited and the necessary data for deep learning models may be generated using FE simulations as was done in this study and other studies (Wu et al [13], Zhan et al [14]). This study shows that deep learning approach can be used for predicting head kinematics from crash videos but needs to be further evaluated on its use on real world data with deep learning models for real-world data developed using either real-world datasets (e.g. Data from NHTSA crash tests) or using simulation datasets with domain adaptation.

While we developed deep learning models for predicting angular velocity time histories, the approach is not limited to angular velocities. It can be used for predicting other kinematic parameters, such as linear accelerations, angular accelerations, etc. as long as video data is available for the subject of interest. The video data in this case become equivalent to the sensors data.

Conclusions

- Deep learning models developed in this study to predict head angular velocities from crash videos show promising results. The models were capable of predicting both the angular velocity time histories and their respective peaks.
- Peak values for ω_z were predicted better than those for ω_y and ω_x . Predictions from these models may be further improved with more training data.
- Deep learning models that can predict kinematics directly from videos provide an alternative method of computing kinematics and can be used:
 - to validate sensor output
 - to cover for faulty sensors
 - to obtain kinematics in the absence of sensors or where head is instrumented with fewer sensors
 - to obtain 3D kinematics from 2D views.

Future Work

Future work involves extending this deep learning concept to physical crash tests to predict ATD head kinematics without the use of sensors.

Declarations

Author contributions statement

V.H. conception, data generation, data preparation, data analysis, deep learning model development and training, writing, editing; E.T. conception, writing, editing.

Competing interests

References

1. Centers for Disease Control and Prevention. *Surveillance Report of Traumatic Brain Injury-related Emergency Department Visits, Hospitalizations, and Deaths—United States, 2014*, U.S. Department of Health and Human Services (2019)
2. Coronado, M. G. & Sugerman, F. Pearson. *The Epidemiology and Prevention of TBI* (2012)
3. Eppinger, R. *et al. Development of Improved Injury Criteria for the Assessment of Advanced Automotive Restraint Systems - II* (NHTSA, USA, 1999).
4. Takhounts, E. G., Craig, M. J., Moorhouse, K., McFadden, J. & Hasija, V. Development of brain injury criteria (BrIC). *Stapp Car Crash J*, **57**, 243–266 (2013).
5. Kimpara, H. & Iwamoto, M. Mild traumatic brain injury predictors based on angular accelerations during impacts. *Ann. Biomed. Eng*, **40**, 114–126 (2012).
6. *Title 49 Code of Federal Regulations (CFR) Part 571 Section 208, Occupant Crash Protection*, National Highway Traffic Safety Administration Washington, DC: Office of the Federal Register, National Archives and Records Administration
7. *The New Car Assessment Program Suggested Approaches for Future Program Enhancements*, DOT HS810698 (2007)
8. Padgaonkar, A. J., Krieger, K. W. & King, A. I. Measurement of angular acceleration of a rigid body using linear accelerometers. *Journal of Applied Mechanics*, pp. 552–556 (1975)
9. Kang, Y., Moorhouse, K. & Bolte, J. H. Measurement of six degrees of freedom head kinematics in impact conditions employing six accelerometers and three angular rate sensors (6aw configuration). *J Biomech Eng* Nov 133(11) (2011)
10. Hollowell, W. T., Gabler, H. C., Stucki, S. L., Summers, S. & Hackney, J. R. *Review of Potential Test Procedures for FMVSS No. 208* (1998)
11. Bartsch, A., Samorezov, S., Benzel, E., Miele, V. & Brett, D. Theoretical development and benchtop, in vitro and in vivo validation of an “Intelligent Mouthguard” (IMG) single event head impact dosimeter, *Stapp Car Crash Journal* (2014)
12. Yoganandan *et al.* Head linear and rotational accelerations and craniocervical loads in lateral impact. *IRCOBI* (2006)
13. Wu, S., Zhao, W., Ghazi, K. & Ji, S. Convolutional neural network for efficient estimation of regional brain strains. *Scientific Reports* volume 9, Article number: 17326 (2019)
14. Zhan, X. *et al.* Deep Learning Head Model for Real-time Estimation of Entire Brain Deformation in Concussion. *ArXiv, abs/2010.08527* (2020)
15. Global Human Body Modeling Consortium (GHBMC), M50-OS, v2.2, (2019)
16. Schwartz, D., Guleyupoglu, B., Koya, B. & *et al.*, “Development of a Computationally Efficient Full Human Body Finite Element Model,” *Traffic Inj. Prev*, **16** (Suppl 1), S49–S56 (2015).

17. Global Human Body Modeling Consortium (GHBMC), M05-OS, v2.2, (2019)
18. Davis, M. L., Koya, B., Schap, J. M. & Gayzik, F. S. Development and Full Body Validation of a 5th Percentile Female Finite Element Model. *Stapp Car Crash J.* Nov; 60:509-544. PMID: 27871105 (2016)
19. Singh, H., Ganesan, V., Davies, J., Paramasuwoom, M. & Gradischnig *Vehicle interior and restraints modeling development of full vehicle finite element model including vehicle interior and occupant restraints systems for occupant safety analysis using THOR dummies (Report No. DOT HS 812 545)* (National Highway Traffic Safety Administration, Washington, DC, 2018).
20. Bradski, G. The OpenCV Library. Dr Dobb's Journal of Software Tools (2000)
21. Yann, L. & Bengio, Yoshua, B. "Convolutional networks for images, speech, and time series". In Arbib, Michael A. (ed.). The handbook of brain theory and neural networks (Second ed.). The MIT press. pp. 276–278 (1995)
22. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput*, **9** (8), 1735-1780 <https://doi.org/doi:10.1162/neco.1997.9.8.1735> (1997). PMID 9377276. S2CID 1915014 (
23. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition, *ICLR* (2015)
24. Lofte, S., Szegedy, C. B. & Normalization Accelerating Deep Network Training by Reducing Internal Covariate Shift, *International Conference on Machine Learning* (2015)
25. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *In NIPS*, pp.1106–1114(2012)
26. Tensorflow v2.4 (2020)
27. Srivastava, N. *et al.* Dropout: A simple way to prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* (2014)
28. Kingma, D. P. & Ba, J. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014)
29. He *et al.* Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, IEEE International Conference on Computer Vision (ICCV) (2015)
30. O'Malley, T., Bursztein, E., Long, J. & Chollet, F. Keras-Tuner (2019)
31. Snoek, J., Larochelle, H. & Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms, *Advances in Neural Information Processing Systems* (2012)
32. Gehre, C., Gades, H. & Wernicke, P. Objective Rating of Signals using Test and Simulation responses. *Enhanced Safety of Vehicles* (2009)
33. Ji, S. *et al.* Group-wise evaluation and comparison of white matter fiber strain and maximum principal strain in sports-related concussion. *J. Neurotrauma*, **32**, 441–454 (2015).
34. Kleiven, S. Predictors for traumatic brain injuries evaluated through accident reconstructions. No. 2007-22-0003. *SAE Technical Paper* (2007)

- 35. He, K., Zhang, X. & Ren, S. Sun,J. Deep Residual Learning for Image Recognition, *Computer Vision and Pattern Recognition* (2015)
- 36. EfficientNet Rethinking Model Scaling for Convolutional Neural Networks, *International Conference on Machine Learning* (2019)
- 37. Ganin *et al.* Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research* vol, 17, 1–35 (2016).

Table 3

Table 3 is available in the Supplementary Files section.

Figures

Figure 1

GHBMC human models in driving position a) 5th female, b) 50th male

Figure 2

Crash pulse used for simulations

Figure 3

Views used to generate videos a) left isometric, b) back isometric, c) front isometric, and d) left

Figure 4

Head motion extracted as a sequence of images

Figure 5

a) Sample input and b) corresponding "ground truth" output for training

Figure 6

a) Full dataset b) Training, Validation and Test datasets

Figure 7

CNN-LSTM Network diagram.

Figure 8

Combined Deep Learning Model

Figure 9

Actual and Predicted time histories for ω_x for 5 random cases from the test dataset

Figure 10

Actual and Predicted time histories for ω_y for 5 random cases from the test dataset

Figure 11

Actual and Predicted time histories for ω_z for 5 random cases from the test dataset

Figure 12

Correlation plots for ω_x , ω_y , and ω_z

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Table3.docx](#)