# Machine Learning for Per-Scene-Encoding

1st Vinzenz Jakob Benedikt Franke
*Information Systems Management*
*Technische Universität Berlin*
Berlin, Germany
vinzenz.franke@campus.tu-berlin.de

2nd Ruihan Zhang
*Wirtschaftsingenieurwesen*
*Technische Universität Berlin*
Berlin, Germany
ruihan.zhang@campus.tu-berlin.de

3rd Julio Cesar Perez Duran
*Computer Science (Informatik)*
*Technische Universität Berlin*
Berlin, Germany
julio.cesar.perez.duran@campus.tu-berlin.de

***Abstract*—The main concept of per-scene encoding is adjusting the bitrate stream to the smallest bitrate which is needed to maintain the high visual quality of each clip in the video to reduce bandwidth consumption. The proposed method to improve conventional per-scene encoding is using machine learning models. In this project, we train and evaluate three machine learning models, which are Linear Regression, Gradient Boosting Decision Tree, and Convolutional Neural Networks. As the result, the encoding ladders for 54 videos were obtained, and the finding that the Gradient Boosting Decision Tree model outperforms the other two models.**

***Index Terms*—per-scene encoding, machine learning, Linear Regression, Gradient Boosting Decision Tree, Convolutional Neural Networks**

## I. Introduction

Video Technology has made huge evolution steps. Since digitization, it is common to stream your favorite movie, instead of buying or renting a DVD at your local store. This was made possible through video encoding. Video Encoding is preparing video for output by compression and decompression [1]. Depending on the Bitrate the user will get a different resolution. Usually the higher the Bitrate, the better the resolution. The increase in bitrate availability led to a yearly rise in global video trafficking.

In Video Encoding it was a common practice to apply the same encoding ladder to all videos, regardless of the content. The best example would be the one-size-fits-all encoding Ladder from Netflix. After that, per-title encoding was introduced to produce one encoding ladder per title. In this paper, we will approach this topic with newer per-scene-encoding, so there would be several encoding ladders for, say a movie, instead of only one.

This paper aims to give improve conventional per-scene encoding with Machine Learning. This paper tests some Machine Learning Algorithm to enhance the VMAF (Video Multimethod Assessment Fusion) [2] quality while avoiding computationally heavy test encodes. In the conducted project, we have built the Machine Learning Algorithm: "Linear Regression", "Convolutional Neural Networks" and "Gradient Boosting Decision Tree". For building, we have used the Metrics MAE (Mean Absolute Error) and VMAF. The Convex Hull was plotted to create the per-scene-encoding Ladder.

This paper is structured as follows. After this introduction, we will provide some related research about Deep Encode Metrics. In Section III our approach for the implementation will be covered, especially the DataSet itself, the Data Analysis, and our different machine learning approaches will be explained more in particular. Section IV represents the Evaluation and Discussion part. The final section V will conclude this paper and the whole project itself.

## II. Related Work

Hundreds of thousands of images shape a two-hour video, say a movie, on average [3]. If users had to receive these massive files without any compression, video streaming would not be possible. Consequently, many efforts to compress these files have come to life assessing how sensitive the human eye is to changes, the way they could take advantage of the redundancy and predictions between frames. Moreover, the interlaced technique came from these efforts, and later more advanced codecs emerged that use lossy compression techniques to deliver videos without a loss of perceptible quality, such as the H.264 [3].

A metric for video quality was necessary to assess how well every codec was delivering a video. Therefore, Netflix, one of the leading enterprises in media streaming, introduced the Video Multimethod Assessment Fusion (VMAF) [2]. , which combines different methods for assessing video quality to deliver a metric that comes close to the perceived quality by the user, therefore a more suitable metric for videos than previous ones such as the PSNR [4].

The bandwidth availability for every user differs; ergo, not all users can receive the same quality. If a user with 1MBPS streams a 4k video, it will probably lead to buffering and stalls that will decrease the perceived quality. Furthermore, receiving the same video with less resolution but without these buffering and stalls would increase the perceived quality. Accordingly, Netflix conducted studies to construct an encoding ladder, a table stating bitrate-resolution pairs to deliver [5].

Not all types of content need the same bitrate to achieve the same quality. Streaming services provide cartoons and sports in full HD, but animation typically requires much less bitrate to achieve it. A sports event can also suffer from stalls and buffering using the one size fits all encoding ladder because of the complexity of the content [6]. This issue motivated Netflix to produce a specialized encoding ladder for every type of content, the per-title approach [6], being able to deliver better quality while using less bandwidth.

It is necessary to test with many encodes for every scene to get the optimal bitrate-resolution pair for each bitrate range, being a very computationally heavy task [7]. As a result, The business unit Future Applications and Media (FAME) at the Fraunhofer Society proposed introducing Artificial Intelligence into the process to avoid this computationally heavy task, and the result was Deep Encode [8]. The idea is to extract information about the video and use a regression algorithm to predict an optimized encoding ladder for every scene. The purpose of this paper is to explain the results of using different regression algorithms and introduce new insights into the scope of regression techniques that Deep

Encode can use to achieve the optimal encoding ladder for every scene.

## III. Approach

### A. Dataset

This project is based on a data set containing 3851 data points containing the encoding and quality information of 54 videos provided by Fraunhofer. There are 3 data categories - source video with the prefix ”s_”, video characteristic with prefix ”c_” and encoding setting with prefix ”e_”. Each data point consists of 46 different features, which mainly represent the video id, the resolution of the raw video, the total size of the source video without audio tracks, the length of the source video, the scan type, the content category, the number of scene changes appearing per minute on average in the video for a given probability, the number of scene changes appearing throughout the entire clip, the spatial perceptual information, the temporal perceptual information, the mean and standard deviation of RGB color values in different blocks, constant rate factor for this encoding, target resolution of the video, the aspect ratio of the video, aspect ratio of the pixels, video codec, video codec profile, video codec level, frames per second, number of frames per I-frames and per interval, the reference frames, scan type as encoding setting, bit depth, color models category, average bitrate as encoding setting and the different types of quality metrics. In brief, the data features include these aspects: the basic information of the source video, the characteristics of the video, the information about encoding settings, and the target quality metric VMAF and the data types are numerical or categorical.

### B. Data analysis

Having the basic knowledge of the dataset, the problem of missing values of the data was investigated. After going through the whole data set, it is visible that the features about the number of scene changes appearing throughout the entire clip, the VMAF for mobile and 4K and PSNR contain only null values. The features about the temporal standard deviation of the mean of the population mean of RGB color values in different blocks contain approximately 80 % null values. Therefore, these data features were not considered. For data with few unknown values, we ignored the unknown variables in our study, e.g. codec profile, scan type.

There are many data features containing constant values, e.g. scan type of source video, the height of source video, and aspect ratio, which can be considered as unimportant features for this project, since they do not provide decisive information about the data.

The impact of data features on the quality metric VMAF was addressed. For further analysis, the verification of the data types is done and then transform all the categorical data, e.g. scan type and video codec, to numeric data to enable the correlation research. Using prepared data, we get the correlation matrix of the data features and concentrate on the correlation with VMAF. We find the data features having the highest absolute correlation coefficients with average VMAF are e_crf with -0.838396, t_average_bitrate with 0.662227, e_width with 0.444841, e_height of 0.444841, e_codec_profile with 0.302228 and e_codec_level with 0.313535. The mentioned absolute correlation coefficients are all higher than 0.3, which shows a strong correlation. Therefore, these features were considered important for the prediction of VMAF values.

### C. Machine learning approaches

The main idea of per scene encoding is to reduce bandwidth consumption by adjusting the bitrate stream to the minimum bitrate which is required to preserve the perfect visual quality of each segment in the video. The entire workflow of encoding is complicated. Improving conventional per-scene encoding with Machine Learning was considered. The acquired encodes were sent to a few machine learning models, which then output an optimal coding ladder. In this project, we chose 3 different machine learning models for per-scene encoding, which are linear regression, Gradient Boosting Decision Tree, and Convolutional Neural Networks. We use VMAF to evaluate the visual quality of the video and predict VMAF values using the mentioned models to build our optimal encoding ladders for each video.

*1) Linear Regression:* Linear Regression is a linear model for modeling the relationship between variables [9]. It is a predictive model, therefore the goal is to predict and forecast different responses.

The first requirement for a Linear Regression Model is the existence of a data set. After data cleaning, it should be tested how high the different variables correlate towards the one variable response, in our case, that would be the VMAF. After selecting the variables with the highest correlation (e.g. Top Five), the data set gets divided into two parts. This happens with a seed, which coincidentally separates the data set. 80 percent of the random data are going into the training data set, while the other 20 percent proceed into the test data set. Now the model can be trained with the training set. All the chosen variables are getting put into their model, the model with the best mean absolute error is chosen. In the next round, the remaining correlating variables get paired with the ”winning” model into a new model. Now again the best model with the best Mean Absolute Error gets selected. This process is repeated, till the mean absolute error doesn’t improve, or every Variable is used in the final model. To assure that there is no over-fitting, the test data set gets tested with the final model from the training data set. If the numbers are quite similar, there should not be over-fitting.

After finishing the model, it is now possible to predict the variable we chose for the correlation analysis (VMAF). This has to be done to get the convex hull and the encoding ladder for every video scene, which contains out of the different bitrate and resolution pairs.

*2) Gradient Boosting Decision Tree:* Gradient boosting decision tree (GBDT) is an iterative decision tree algorithm that comprises multiple decision trees where the conclusions of all trees are accumulated to form the final answer. GBDT can be used for almost all regression problems (linear/nonlinear) and has very wide applicability.

It is important to note that the decision tree here is a regression tree. Each node of the regression tree is given a prediction value, which, with bitrate, is equal to the average of the bitrates of all data points belonging to the node. The branching process exhausts each threshold of each feature to find the best split, and the best measure is the minimization of the squared error. The more data points are predicted to be wrong, the more outrageous the predictions are, the higher the squared error is, and the most reliable branching can be found by minimizing the squared error. Branching obtains two new nodes according to this criterion. The branching is done until the bitrate of data points on each leaf node is unique or reaches a preset termination condition (e.g., the upper limit of the number of leaves), and if the bitrate of data points on the final leaf node is not unique, the average bitrate of all data points on that node is used as the predicted bitrate of that leaf node.

Boosting trees are iterative multiple regression trees that decide together. When the squared error loss function is used, each regression tree learns the conclusions and residuals of all previous trees and fits to get a current residual regression tree with the meaning of residuals as in the formula: residuals = true value - predicted value. A boosting tree is the accumulation of regression trees generated by the whole iteration process.

Gradient boosting is a boosting method. Its main idea is that each time the model is built is toward the gradient descent of the loss function of the previously built model, assuming that the current model under the stepwise model is f(x), the value of the negative gradient of the loss function L under f(x) is used as the residual in the boosting tree algorithm to fit a regression tree.

In this project, the input variables used for GBDT model are ”s_video_id”, ”e_crf”, ”e_width”, ”e_height”, ”e_codec_profile”, ”e_codec_level” and ”t_average_bitrate” and the predicted variable is VMAF. The input variable ”s_video_id” is selected since we need to build encoding ladders for each video. And the other input variables are the features which have a high correlation with VMAF as mentioned in 3.B.

It is important to select parameters for the GBDT model. The used parameters in our model are max_depth, n_estimators, and learning_rate. Max_depth refers to the number of leaves of each tree and n_estimators refers to the total number of trees in the ensemble. To tune the parameters, we can plot them concerning the scores using a cleaned dataset and select parameters achieving higher scores within the range. When selecting parameters, it is often necessary to pay attention to the interaction between the learning rate and the number of iterations. Reducing the learning rate and increasing the number of iterations within a certain range usually can increase fit while maintaining the generalization ability. Finally, max_depth equal to 4, n_estimators equal to 500, and learning_rate equal to 0.1 are selected in this model.

*3) Convolutional Neural Networks:* A convolutional neural network is a deep neural network with at least one convolutional layer. In this layer take place the convolutions, linear operations involving the multiplication of an array of data and a two-dimensional array of weights [10].

This machine learning model is suitable for images. Images or frames define videos, and the data comes from videos. As a result, it is also appropriate for videos. In addition, it also supports regression.

Three convolutional layers, two hidden layers, a max-pooling layer, and a flatten layer shape the model. In addition, it leverages a dropout layer to prevent the over-fitting of the neural network. It takes as input: e_crf, t_average_bitrate, e_height, e_width, e_scan_type, e_codec level, e_codec,_profile,c_si and has the predicted VMAF as the output.

## IV. Evaluation / Discussion

### A. Lineare Regression

The MAE (Mean Absolute Error) was used to evaluating the Linear Regression Model. MAE is the average difference between the predicted linear regression model VMAF values and the true VMAF value. As a decisive factor, the sMAPE (symmetric Mean Absolute Percentage Error) is used as well. For the training set, the MAE for the best and final model was 7.1028, and the sMAPE had the value of 0.3504. For the test set the MAE for the model was 7.0957 and for the sMAPE 0.3377, so slightly better than the training set, but still close enough to avoid over-fitting. The final variables used for the model were: ”e_crf”, ”e_width”, ”e_codec_profileNumber”, ”e_codec_level”, ”e_scan_typeNumber” and ”e_height”. In R Studio it is not possible to plot the 2 graphs in one graph. You can still clearly see the different VMAF performances for the different resolutions in “Fig. 1”.

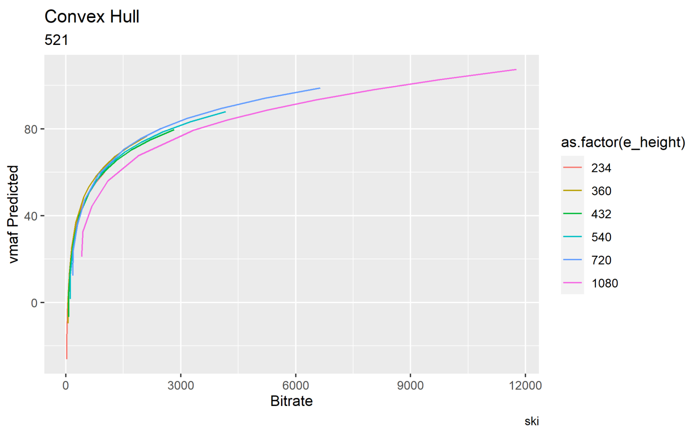


Fig. 1. Bitrate/VMAF graph for video 521.

In “Fig. 2” the encoding Ladder is shown.

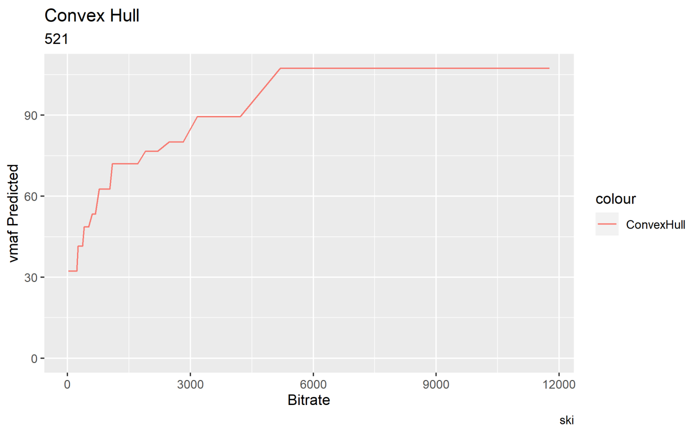


Fig. 2. Bitrate/VMAF graph for video 521.

The encoding ladder contains the videoID, the resolution, bitrate, and the predicted highest VMAF for that pair. As it is shown for the videoID 521 in “Fig. 3”

In “Fig. 4” and “Fig. 5” there is the convex Hull for video ID 529, which had no content category, as you can see, the graphs are not too different. It is the same with the Encoding Ladder, which you can see in “Fig. 6”.

| ï..s_video_id | bitrate | resolution | maxVMAF |
|---|---|---|---|
| 521 | 235 | 1280x720 | 32.21934 |
| 521 | 375 | 1280x720 | 41.55553 |
| 521 | 560 | 1920x1080 | 48.66075 |
| 521 | 750 | 1920x1080 | 53.32884 |
| 521 | 1050 | 1280x720 | 62.66503 |
| 521 | 1750 | 1920x1080 | 72.00121 |
| 521 | 2350 | 1920x1080 | 76.66930 |
| 521 | 3000 | 1280x720 | 80.11126 |
| 521 | 4300 | 1920x1080 | 89.44744 |
| 521 | 5800 | 1920x1080 | 107.37352 |

Fig. 3. Encoding Ladder graph for video 521.

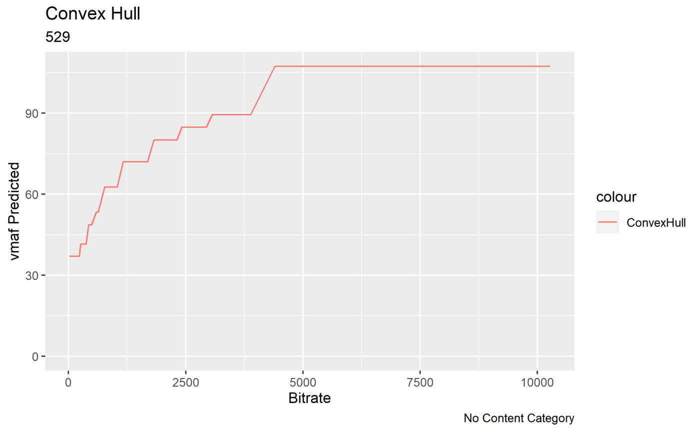


Fig. 4. Bitrate/VMAF graph for video 529.

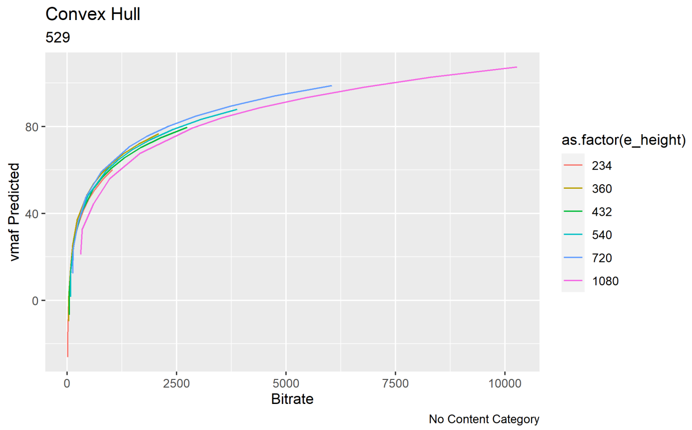


Fig. 5. Bitrate/VMAF graph for video 529.

| ï..s_video_id | bitrate | resolution | maxVMAF |
|---|---|---|---|
| 529 | 375 | 1920x1080 | 41.55553 |
| 529 | 560 | 1280x720 | 48.66075 |
| 529 | 750 | 1920x1080 | 53.32884 |
| 529 | 1050 | 1920x1080 | 62.66503 |
| 529 | 1750 | 1920x1080 | 72.00121 |
| 529 | 2350 | 1280x720 | 80.11126 |
| 529 | 3000 | 1920x1080 | 84.77935 |
| 529 | 4300 | 1920x1080 | 89.44744 |
| 529 | 5800 | 1920x1080 | 107.37352 |

Fig. 6. Encoding Ladder graph for video 529.

### *B. Gradient Boosting Decision Tree*

To evaluate the GBDT model, the Mean Absolute Error (MAE) and the score are used. The mean absolute error is the average of the distance between the model prediction VMAF value and the sample true VMAF value. It is worth mentioning that one advantage of MAE over Mean Squared Error (MSE) is that MAE is less sensitive to outliers and more tolerant. The score in this model is the coefficient of determination of the prediction. The coefficient of determination $R^2$ is defined as $(1 - \frac{u}{v})$, where u is the residual sum of squares $\sum$ (true VMAF - predicted VMAF)$^2$ and v is the total sum of squares $\sum$ (true VMAF - mean of true VMAF)$^2$. The best possible score is 1.0. The MAE of the test set in this model is 1.5429 and the score of the test set is 0.99495. It shows that this model has high fitting and has no over-fitting problem because what we evaluate is the test set.

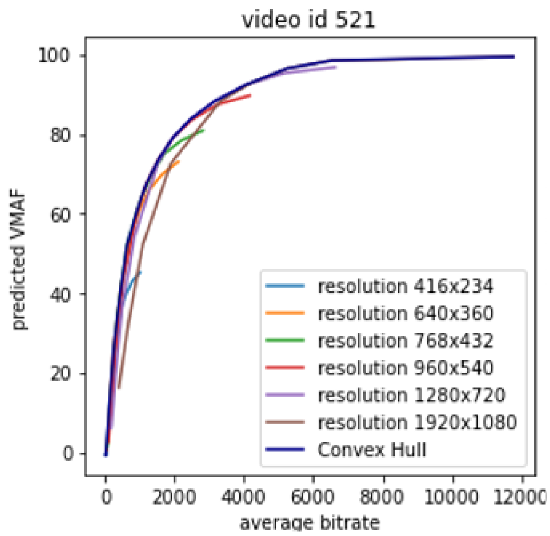


Fig. 7. Bitrate/VMAF graph for video 521.

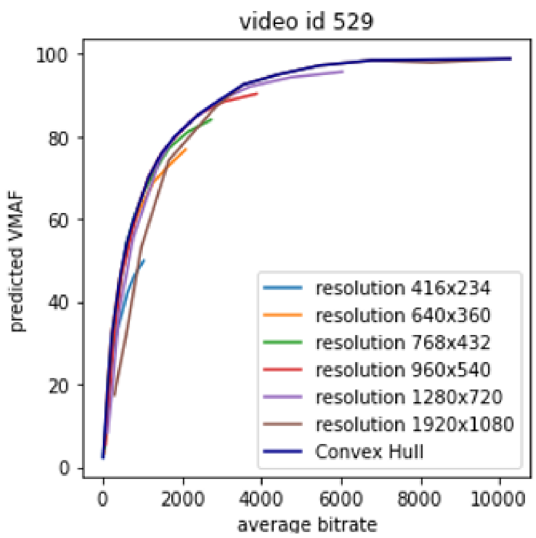


Fig. 8. Bitrate/VMAF graph for video 529.

From Fig. 7 and Fig. 8, a better understanding of the relation between bitrate and VMAF values can be obtained. The x-axis represents the bitrate, while the y-axis shows the predicted VMAF value. The labels with different colors are different resolutions. It is clear to see that some curves intersect in this graph, which shows that with the same bitrate the VMAF value by lower resolution is higher sometimes. Therefore, it is meaningful to find the convex hull of these curves, which is helpful to save the bitrate and build the encoding ladder. The dark blue curve in this graph is the convex hull, which shows the maximal VMAF values for every bitrate.

With the help of the convex hull, the encoding ladder can be built, which is useful to save bitrate and achieve higher VMAF values. As an example, we can see Fig. 9 the encoding ladder for video ID 521 and Fig. 10 the encoding ladder for video ID 529, which contain the resolution, the bitrate, and predicted VMAF values.

| Resolution | Bitrate | VMAF(predicted) |
|---|---|---|
| 416x234 | 123 | 11 |
| 640x360 | 470 | 42 |
| 768x432 | 804 | 57 |
| 960x540 | 1201 | 68 |
| 1280x720 | 1944 | 79 |

Fig. 9. The Encoding ladder for video 521.

| Resolution | Bitrate | VMAF(predicted) |
|---|---|---|
| 640x360 | 456 | 47 |
| 768x432 | 791 | 60 |
| 960x540 | 1162 | 70 |
| 1280x720 | 1822 | 80 |

Fig. 10. The Encoding ladder for video 529.

## C. Convolutional Neural Network

The Mean Absolute Error is a metric that can show the difference between the predicted VMAF values and the real ones [11]. One training could lead to biases since this model depends on initial random variables. The result of 200 randomly split tests was 2.969299849271774 with a standard error of 0.01193681736466972. Each time 80 % of the data went for the training of the model and the rest to measure the accuracy.

By plotting the results of predicting the VMAF values into a graph, it is clear how some resolutions perform better than others at certain bitrates. The example graphs “Fig. 11” and “Fig. 12” show this phenomenon. The x-axis corresponds to the bitrate, and the y-axis represents the predicted VMAF. Every color indicates a different pixel resolution. Each graph matches one video. Accordingly, the Convex Hull was also highlighted with green to represent the ideal resolution for every bitrate to get the maximum VMAF. The spline method smoothed the curves and got more data points for the graph. Combining this calculation with the bitrate ranges of the traditional one-size-fits-all ladder of Netflix results in the encoding ladders as in “Fig. 13” and “Fig. 14”.

## D. Comparison/Discussion

In Comparison, the Gradient Boosting Decision Tree showed the promising values in terms of accuracy; it had the best MAE with a rounded value of 1.54, the Convolutional Neural Networks was close behind with a rounded value of 2.97, the worst of the three machine learning algorithms was the linear regression model with a rounded value of

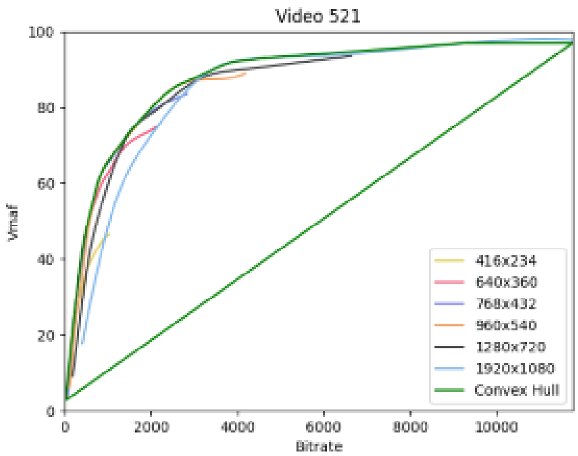


Fig. 11. The graph for the video 521.

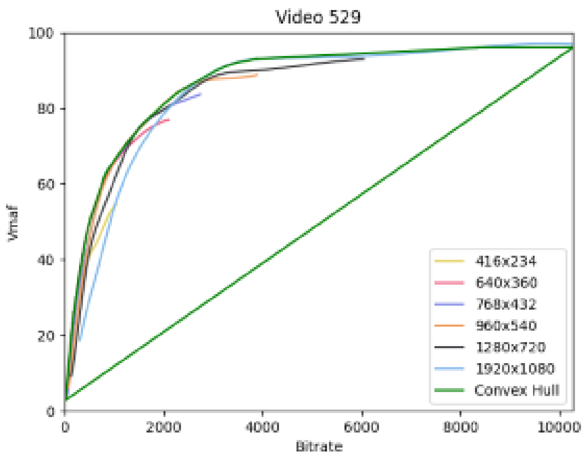


Fig. 12. The graph for the video 529.

Video 521

| Bitrate (kbps) | Resolution | VMAF (Predicted) |
|---|---|---|
| 235 | 768x432 | 35.685078 |
| 375 | 640x360 | 46.575527 |
| 560 | 768x432 | 53.59677 |
| 750 | 768x432 | 65.7068 |
| 1050 | 768x432 | 76.12955 |
| 1750 | 768x432 | 80.317314 |
| 2350 | 960x540 | 85.26026 |
| 3000 | 1920x1080 | 92.60407 |
| 4300 | 1920x1080 | 93.325264 |
| 5800 | 1920x1080 | 97.8191 |

Fig. 13. The encoding ladder for the video 521.

| Bitrate (kbps) | Resolution | VMAF (Predicted) |
|---|---|---|
| 235 | 416x234 | 35.478123 |
| 375 | 640x360 | 47.643276 |
| 560 | 640x360 | 53.71951 |
| 750 | 768x432 | 66.30605 |
| 1050 | 768x432 | 76.76008 |
| 1750 | 1280x720 | 82.45278 |
| 2350 | 1920x1080 | 88.59334 |
| 3000 | 1920x1080 | 92.33311 |
| 4300 | 1920x1080 | 93.50517 |
| 5800 | 1920x1080 | 96.83489 |

Fig. 14. The encoding ladder for the video 529.

7.10. The final results were for all three machine learning algorithms good, but for the Gradient Boosting Decision Tree and Convolutional Neural Networks exceptional good. The Encoding Ladder for the three different Algorithms is quite similar, especially the linear regression model and the convolutional neural networks show a similar result.

To improve conventional per-scene encoding the machine learning algorithms Convolutional Neural Networks and Gradient Boosting Decision Tree are recommended to use.

## V. Conclusion

Deep Encode is a project that takes the per-scene encoding approach further by leveraging Machine Learning to avoid the need for computationally heavy test encodes [8]. It extracts information from the videos and selects the most suitable regression algorithm to predict an optimized encoding ladder for each video. This work adds regression techniques to the selection of algorithms Deep Encode can use while providing insights to choose the best one according to the data of the video. Furthermore, the predictions of the machine learning algorithms are very close to reality, taking into account that the maximum mean absolute error does not surpass 8 points in any case. With the lowest MAE, the Gradient Boosting Decision Tree method is singularly accurate in this study, which means that with data similar to the one used, this should be the selected algorithm. Deep Encode could benefit considerably from adding this algorithm to its selection, enhancing the accuracy of the encoding ladder.

The future of streaming can benefit substantially from Deep Encode and slowly progress into a response closer to real-time for users while at the same time providing better quality with lower bandwidth and reducing the computational complexity. However, the data used, which belongs to 54 scenes, does not represent the wide variety of content that traditional streaming services such as Netflix offer. Therefore, a more thorough study using the Gradient Boosting Decision Tree is necessary to evaluate the performance in a real-world situation.

## References


[1] C. A. L. Taha, M., “A qoe adaptive management system for high definition video streaming over wireless networks,” *Telecommun Syst 77*, vol. 77, 2021.

[2] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward a practical perceptual video quality metric,” *The Netflix Tech Blog*, vol. 6, no. 2, 2016.

[3] L. Moreira. (2021) Digital video introduction. [Online]. Available: https://github.com/leandromoreira/digital_video_introduction

[4] A. Horé and D. Ziou, “Image quality metrics: Psnr vs. ssim,” in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.

[5] A. Aaron, Z. Li, M. Manohara, J. D. Cock, and D. Ronca. (2015) Per-title encode optimization. [Online]. Available: https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2

[6] J. De Cock, Z. Li, M. Manohara, and A. Aaron, “Complexity-based consistent-quality encoding in the cloud,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1484–1488.

[7] D. Silhavy. (2019) Per-title encoding. [Online]. Available: https://websites.fraunhofer.de/video-dev/per-title-encoding/

[8] Famium deep encode. [Online]. Available: https://www.fokus.fraunhofer.de/en/fame/deep-encode

[9] O. O. Allen, “A linear regression model for the analysis of life times,” *Statistics in Medicine*, vol. 8, 1989.

[10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[11] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance,” *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.