A Vision-based Solution for Track Misalignment Detection

Koteswar Rao Jerripothula*
Indraprastha Institute of Information Technology Delhi
New Delhi, India
koteswar@iiitd.ac.in

Sharik Ali Ansari*
COER
Roorkee, India
kunwar.sharik@gmail.com

Rahul Nijhawan
UPES
Dehradun, India
rahulnijhawan2010@gmail.com

Abstract—**Derailment is one of the most frequent ways railway accidents happen. Track defects such as buckling and hogging that cause misalignment of tracks can easily lead to derailments. While railway tracks get laterally misaligned due to buckling, vertical misalignments can result from hogging. Such misalignments are visibly recognizable, and we can even automate recognition using data-driven models. This paper discusses how we build such data-driven models. There are no public datasets available to build such models; therefore, we introduce TMD (Track Misalignment Detection) dataset. It consists of misaligned and normal track images. The problem we try to solve here is essentially a binary image classification problem, which we solve by exploring the feature extraction approach to transfer learning (TL). In this approach, we employ a pre-trained network to extract rich features, which are then supplied with annotations to a learning algorithm for building a candidate TL model. Several pre-trained networks and learning algorithms exist, resulting in multiple candidate TL models; therefore, it becomes essential to identify effective ones. We propose an evaluation criterion to decide which are effective ones using our proposed bias-variance analysis. Our experiments demonstrate that the candidate TL models selected based on our proposed evaluation criterion perform better than other candidate TL models while testing.**

I. Introduction

Derailment is one of the severe problems that the railway mode of transportation has kept on facing since its inception. This problem has caused many deaths, injuries, and financial losses. According to the Association of American Railroads, train accidents are caused by defective tracks, faulty pieces of equipment, and human errors. Defects in railway tracks are of two types: (i) ones that are visible from a far distance, misalignments; and (ii) ones that need a closer inspection, such as corrugation of rails, wheel burns, wear and crack on rails, rusted rails, and loose joints. In this paper, we consider the defects of the first type. While defects like buckling of the tracks lead to lateral misalignment, hogging defects lead to vertical misalignment.

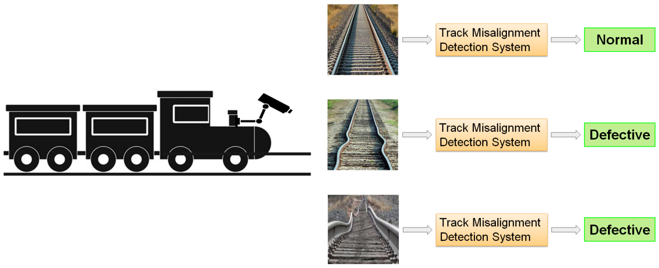

Track Misalignment Detection System
Normal
Track Misalignment Detection System
Defective
Track Misalignment Detection System
Defective

Fig. 1. Our objective is to develop a data-driven solution for detecting track misalignments using a camera mounted in front of trains or railway inspection vehicles. While the first track is normal, the second and the third ones have buckling (horizontal misalignment) and hogging (vertical misalignment) defects, respectively.

We propose a computer vision solution that can help detect such misalignments. It can help automate railway inspections and warn railway drivers of such misalignments from far away in real-time. More specifically, we develop a data-driven solution that can classify the incoming images from a camera (mounted in front of a train or railway inspection vehicle) into defective or normal, as shown in Fig. 1. The tracks having buckling (2^{nd} case) or hogging (3^{rd} case) defects are recognized as defective.

Despite such misalignments being visible from far away, it is quite challenging to model them in an unsupervised manner. For an unsupervised approach, we may have to segment [1], [2] out the tracks and perform shape analysis, which are very challenging tasks in a natural setting. Even in the supervised learning approach, effective feature design/extraction [3] for this problem remains a big concern. Recently, deep learning has received great attention in computer vision research, for it can help computers learn features too. However, taking such an approach from scratch requires large amounts of labeled data to successfully train a deep neural network that can generate rich features and eventually perform the expected predictions.

Interestingly, once a deep neural network is trained, the same network can also help us solve closely related problems through transfer learning. Transfer learning requires far less labeled data than what is needed to learn from scratch. Specifically, we explore the feature extraction approach of transfer learning, where we use pre-trained networks as feature extractors to generate the features required for learning algorithms. However, this approach involves hyperparameters, namely pre-trained networks and learning algorithms. While we vary these hyperparameters, we evaluate the resultant models using an evaluation metric we develop. The main idea is to model bias and variance errors using training and cross-validation

*indicates equal contribution

accuracies and find models with minimum joint error. Since there is no publicly available dataset on this problem, we collected several railway-track images and provided ground-truth labels of whether they are defective or normal tracks. We call it *TMD: Track Misalignment Detection*[1] dataset.

Our main contributions are as follows: (i) We develop a new dataset for track misalignment detection. (ii) This is the first work to develop a vision-based solution for track misalignment detection.

The remaining sections are arranged in the following way. First, we will discuss how existing works detect different track defects. Second, we will discuss the dataset we have built to solve out track misalignment detection problem. Third, we will discuss how we select effective TL models. Lastly, we will discuss our experimental results.

II. Related Works

Development in technology led to the implementation of various techniques to prevent train accidents. A robotic stick, proposed by [4], detects and characterizes rolling contact fatigue cracks using a combined threshold and signature match algorithm. The authors of [5] proposed a 3D laser-based method for detecting abrasion, scratch, and peeling of the rail surface using k-means clustering and the decision tree algorithm. Authors in [6] also detect surface defects using a laser-based 3D model. A different approach that uses Acoustic Emission (AE) was used in [7] to detect rail defects at high speed; they used multivariate acoustic noise cancellation to remove noise from waves at high speed.

There are also some vision-based works. Authors in [8] proposed to utilize gray-level co-occurrence matrix and LBP (Localized Binary Patterns) features to obtain texture features, which can be used to identify whether there is a crack in the track or not. Another method of crack detection was proposed in [9] by comparing the rail image with faulty rail track images. Then, there are railway subgrade defects, which were identified using Feature Cascade, adversarial spatial dropout Network, non-maximum suppression, and R-CNN in [10]. The work proposed in [11] performs real-time detection of rail surface defects at different speeds using the contour of the direction chain code on morphologically processed images. Authors in [12] proposed a method to detect surface defects by first obtaining only the image of the rail, excluding other parts of the image using the Canny edge detector, and then detecting the defect using only the image of the rail using pre-trained CNN. In [13], authors develop a method to enable the positioning and classification of the track and fasteners to detect defects; however, it requires a close view of the individual tracks. Authors of [14] also deal with fastener related defects. Although many vision-based solutions have been proposed, through both hand-crafted features and deep learning, none focus on detecting track misalignments. Our paper proposes a solution to this underexplored problem of detecting misalignments caused by buckling and hogging. Moreover, many methods require images to have a close view of tracks, which we do not. We try to detect defects that are visible from far away.

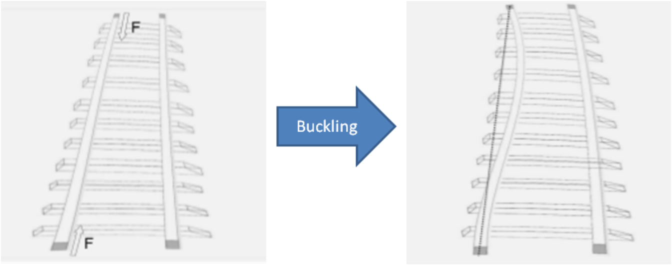


Fig. 2. Buckling Phenomenon: When there is a development of longitudinal compressive stress (represented by F), the track starts buckling from its original position. It is basically characterized by the lateral misalignment of the track.

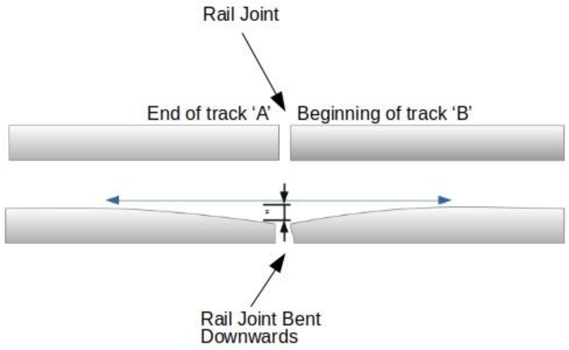


Fig. 3. Hogging Phenomenon: It is basically characterized by the vertical misalignment of the track at the ends. It occurs due to the battering (repeated hitting) action of train wheels.

III. TMD: *Track Misalignment Detection* dataset

Our dataset consists of images of two types of tracks: defective and normal. We especially consider buckling and hogging defects.

Buckling is characterized by lateral misalignment of railway tracks, as illustrated in Fig. 2. It occurs due to the longitudinal compressive stress that builds inside the rail due to the difference in the temperature compared to the RNT (Rail Neutral Temperature). RNT is the temperature at which the rail is in a ”stress-free” state. The weakened condition of the track and the vehicle load also contributes to buckling, according to [15]. When a train passes through such a laterally misaligned track, the train wheel may lose contact with the track, causing a derailment. Reports in [16], [17] show several derailments that have resulted due to buckling.

The other defect, hogging, occurs primarily due to train wheels' battering (repeated hitting) action. It causes tracks to bend down at their ends, where the tracks are joined to form continuity. It results in a dip (a vertical misalignment) at such joints. This phenomenon can be seen in Fig. 3. When one

[1] Our TMD dataset can be downloaded from the following webpage: https://sites.google.com/site/koteswarraojerripothula/neatai-servolab/track-misalignment-detection-sibgrapi-2021

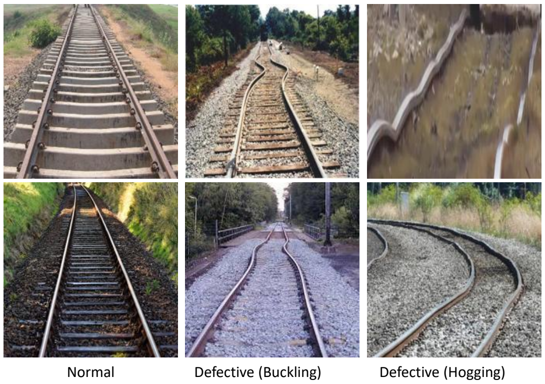


Fig. 4. Sample images from our Track Misalignment Detection (TMD) dataset.

TABLE I
DISTRIBUTION OF IMAGES IN OUR TRACK MISALIGNMENT DETECTION (TMD) DATASET

Train		Test	
Defective	Normal	Defective	Normal
221	223	55	56

such dip occurs in the railway track, the dip causes the train to bounce. Further, when a train bounces, it pushes the track downwards while landing. Due to such pressure, another dip starts to form. In this manner, a sequence of dips is created as the dip causes the train to bounce [18]. The passage of many trains causes the formed dips to get deeper and deeper up to the point where the landing may not be on track, resulting in a derailment. Such a defect is also called cyclic-top due to the formation of dips in a cyclic manner. The reports in [19], [20] discuss several instances of derailments that occurred due to hogging defects in tracks.

We collected a total of 555 track images for our Track Misalignment Detection (TMD) dataset from the Internet. The distribution of images in our dataset is provided in Table I. While we keep 20% of our data for testing, the remaining portion is used for training. We carefully select and crop these images such that they appear to have been taken from a camera mounted on a train engine or a railway inspection vehicle. All the images comprise a single track. We split the images having two tracks into two to have more samples. We show some sample images of normal and defective tracks in Fig. 4. All the images have been resized to 224x224 for consistency. We found some grayscale images too, which we converted to color by simply repeating the image data in all three color channels. Our TMD dataset can be downloaded from the following webpage: https://sites.google.com/site/koteswarraojerripothula/neatai-servolab/track-misalignment-detection-sibgrapi-2021.

IV. Transfer Learning

There are two hyperparameters in the feature extraction approach of transfer learning (TL), namely the feature extractor (pre-trained network) and the learning algorithm. As we vary these TL hyperparameters, we will have several candidate TL models with different degrees of effectiveness. Our goal here is to identify the most effective models before we test them. In this section, we develop a criterion to evaluate these candidate TL models.

Numerous pre-trained networks can take the role of feature extractors, such as Inception v3, VGG16, VGG19, and so on. Let $\mathcal{F} = \{f_1, f_2, \cdots, f_m\}$ denote the set of m feature extractors available for consideration. Similarly, numerous learning algorithms are available for consideration, such as decision trees, SVM, logistic regression, etc. Let $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ denote the set of n learning algorithms we have. In this way, there are m (feature extractors) × n (learning algorithms) TL models possible. Out of these $m \times n$ TL models, some TL models will be very effective. These models should have low bias and variance errors. To quantify this observation, we need to model these errors.

A. Bias Error

A TL model is likely to have high bias error when it is too simple and performs poorly during training and validation. [*Note: We use cross-validation strategy for validation*]. So the error is inversely proportional to these accuracies. If we assume the relationship to be inverse of exponential, for a given f_i and a_j, we can model bias error in the following manner:

$$B(f_i, a_j) \propto e^{-T(f_i,a_j)} e^{-V(f_i,a_j)} \tag{1}$$

where $B(f_i, a_j)$ denotes the bias error and $T(f_i, a_j)$ and $V(f_i, a_j)$ denote the training and validation accuracies, respectively, for the TL models built using f_i and a_j. By modeling bias errors in this way, we can expect its value to be minimum when both the accuracies are high, and maximum when both the accuracies are 0.

B. Variance Error

A TL model will have a high variance error when it is not general and tends to be too specific to the training data. Similar to the bias error discussed above, we again consider training and validation accuracies to model the variance error. A model that generalizes well should get similar accuracies during training and validation. Therefore, the higher the similarity between the two accuracies, the smaller the variance error. Thus, for a given f_i and a_j, we can model variance error (denoted by $S(f_i, a_j)$) in the following way:

$$S(f_i, a_j) \propto e^{(T(f_i,a_j) - V(f_i,a_j))}, \tag{2}$$

which ensures that the value will be minimum only when $T(f_i, a_j) = V(f_i, a_j)$, which means the model has generalized well. In contrast, if there is a high difference, the error will also become high.

C. Joint Error

Both the above discussed errors are important, and they need to be jointly minimized. We combine these two errors

TABLE II
OUR TRAINING ACCURACY $T(\cdot)$, CROSS-VALIDATION ACCURACY $V(\cdot)$ AND EVALUATION METRIC VALUE $X(\cdot)$ OBTAINED FOR DIFFERENT DIFFERENT TL MODELS. FROM THIS TABLE, THE APPROACHES HIGHLIGHTED IN BLUE ARE SELECTED AS THE MOST EFFECTIVE ONES BASED ON OUR EVALUATION CRITERION (7).

$a\backslash f$	Inception v3			SqueezeNet			VGG16			VGG19		
	$T(f,a)$	$V(f,a)$	$X(f,a)$	$T(f,a)$	$V(f,a)$	$X(f,a)$	$T(f,a)$	$V(f,a)$	$X(f,a)$	$T(f,a)$	$V(f,a)$	$X(f,a)$
kNN	0.948	0.890	0.905	0.917	0.867	0.880	0.935	0.892	0.903	0.914	0.874	0.884
DT	0.973	0.739	0.798	0.982	0.840	0.876	0.993	0.824	0.866	0.977	0.840	0.874
SVM	0.993	0.917	0.936	0.991	0.914	0.933	0.980	0.910	0.928	0.986	0.890	0.914
SGD	1.000	0.930	0.948	1.000	0.930	0.948	1.000	0.935	0.951	1.000	0.932	0.949
RF	0.993	0.854	0.889	0.991	0.865	0.897	0.998	0.899	0.924	0.993	0.887	0.914
NN	1.000	0.937	0.953	1.000	0.939	0.954	1.000	0.948	0.961	1.000	0.941	0.956
NB	0.851	0.829	0.835	0.867	0.856	0.859	0.880	0.863	0.867	0.881	0.840	0.850
LR	1.000	0.926	0.945	1.000	0.932	0.949	1.000	0.948	0.961	1.000	0.944	0.958
GB	1.000	0.887	0.915	1.000	0.908	0.931	1.000	0.921	0.941	1.000	0.921	0.941
AB	1.000	0.793	0.845	1.000	0.842	0.882	1.000	0.863	0.897	1.000	0.831	0.873

by multiplying them, as shown below, to obtain what we call joint error $\epsilon(\cdot)$:

$$\epsilon(f_i, a_j) = B(f_i, a_j)^{\alpha} \times S(f_i, a_j)^{\beta} \tag{3}$$

where α and β having 0-1 range represent weights given to the two errors. If this joint error is low, we can consider the TL model to be very effective. Let the proportionality constants for the Eqns.(1) and (2) be ρ_b and ρ_s, respectively. Now, when we substitute the two errors in Eqn.(3), we get the following relationship:

$$\epsilon(f_i, a_j) = \rho_b^{\alpha} \rho_s^{\beta} e^{-\alpha\left(T(f_i,a_j)+V(f_i,a_j)\right)+\beta\left(T(f_i,a_j)-V(f_i,a_j)\right)} \tag{4}$$

Remember that ρ_b,ρ_s, α and β are all constants. Hence, the term $\rho_b^{\alpha}\rho_s^{\beta}$ has no role to play when we try to minimize the error. Hence, only the exponential term remains for minimizing.

D. TL Model Selection

We can achieve our objective of finding the most effective TL models by minimizing the exponential term of Eqn.(4), which means maximizing the negative of its exponent. Hence, this negative of the exponent becomes our evaluation metric. The higher the value, the better. Therefore, let $X(\cdot)$ denote our evaluation metric, which can be expressed as follows, after rearrangement of the terms involved:

$$X(f_i, a_j) = (\alpha + \beta)V(f_i, a_j) + (\alpha - \beta)T(f_i, a_j). \tag{5}$$

Now, if we want to have the 0-1 range for X, just like T and V, we need to have the sum of the coefficients of training and validation accuracies to be one, which results in $\alpha = \frac{1}{2}$. Hence, the final expression of X becomes the following:

$$X(f_i, a_j) = (\frac{1}{2} + \beta)V(f_i, a_j) + (\frac{1}{2} - \beta)T(f_i, a_j), \tag{6}$$

where if we set β also to $\frac{1}{2}$, our evaluation criterion will then just rely on validation accuracy. Sometimes, validation accuracy alone can be misleading, when the number of samples is too small, which can happen in TL. Therefore, we set β as 0.25 as the default value to ensuring that V has 0.75 weightage and T has at least 0.25 weightage.

TABLE III
CLASSIFICATION ACCURACIES OF DIFFERENT TL MODELS ON TEST SET. ONLY 4 (SHOWN IN RED) OUT OF 33 UNSELECTED MODELS COULD CROSS/TOUCH THE AVERAGE OBTAINED BY OUR EFFECTIVE TL MODELS (SHOWN IN BLUE).

	Inceptionv3	SqueezeNet	VGG16	VGG19
kNN	0.838	0.811	0.865	0.829
DT	0.712	0.775	0.838	0.820
SVM	0.910	0.874	0.901	0.910
SGD	0.937	0.892	0.937	0.919
RF	0.820	0.856	0.856	0.847
NN	0.946	0.874	0.928	0.928
NB	0.847	0.793	0.820	0.865
LR	0.946	0.901	0.919	0.901
GB	0.901	0.856	0.928	0.883
AB	0.775	0.775	0.811	0.829

Now that we have an expression for our evaluation metric $X(\cdot)$, we can find effective candidate TL models based on the following criterion:

$$X(f_i, a_j) > (mX + sX) \tag{7}$$

where mX and sX denote average and standard deviation of all the X scores available.

V. EXPERIMENTAL RESULTS

A. Evaluation Metric (X) Scores

As mentioned earlier, we consider two transfer learning hyperparameters: feature extractor and learning algorithm. Specifically, we use Inception v3 [21], VGG-16 [22], VGG-19 [22] and Squeezenet [23] as possible feature extractors, pre-trained on ImageNet. As far as the learning algorithms are concerned, we use Ada Boost (AB), Decision Tree (DT), k-Nearest Neighbors (kNN), Neural Networks (NN), Logistic Regression (LR), Naive Bayes (NB), Gradient Boosting (GB), Random Forests (RF), Stochastic Gradient Descent (SGD) and Support Vectors Machine (SVM) algorithms. Hence, we end up developing $4 \times 10 = 40$ TL models using our training subset, and we need to select the most effective ones based on the criterion given in Eqn.(7). The training and validation accuracies of each candidate TL model is given in Table II. By accuracy, we mean classification accuracy. We also report

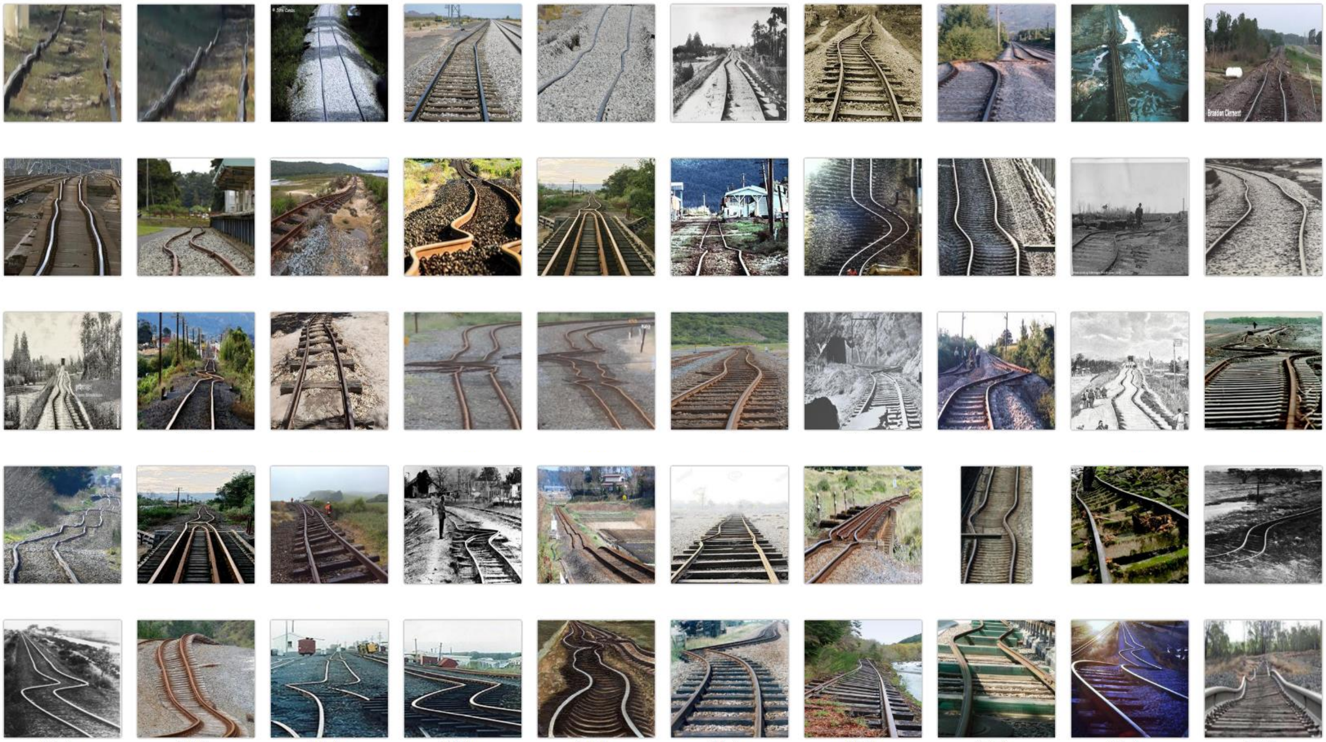


Fig. 5. Sample test images that have been correctly classified as defective by our best TL model InceptionV3+NN.

the corresponding values of our $X(\cdot)$ scores. We obtain the threshold value $mX + sV$ equal to 0.9508 from these scores. Out of these 40 TL models, a total of 7 models satisfy our selection criterion (7) and have been shown in blue. The seven selected TL models are InceptionV3+NN, SqueezeNet+NN, VGG16+SGD, VGG16+NN, VGG16+LR, VGG19+NN and VGG19+LR.

We conduct all our experiments using Python's Orange library. To have reproducible results without much difficulty, since there are 40 TL models, we simply use the default settings of its functions. For example, 'k' in the k-fold cross-validation (used for obtaining validation scores V) is 5. Similarly, 'k' in the kNN learning algorithm is also 5. We encourage readers to refer to Orange's widget catalog[2] to know more about these default settings.

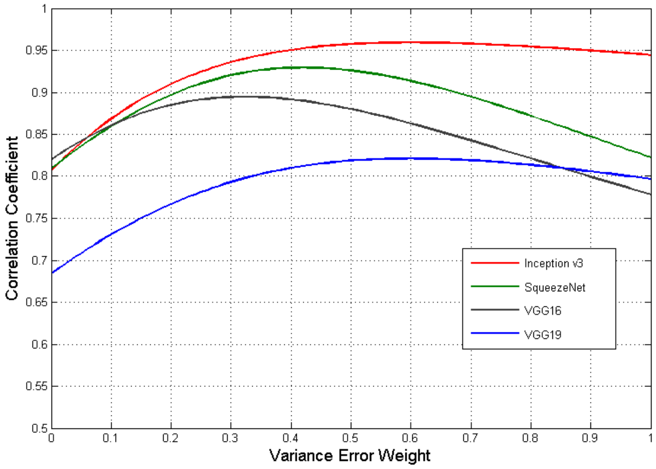


Fig. 6. Correlation Coefficient (between X scores and test results of a feature extractor for different learning algorithms) while varying the variance error weight (β)

B. Test Results

Having obtained the most effective TL models, let's compare them (shown in blue) with other TL models in terms of test accuracies given in Table III. As it can be seen, most of the other TL models have test accuracies less than the average test accuracies of the selected TL models. The average is 0.919, and only 4 (shown in red) out of the remaining 33 TL models could cross/touch that mark. Among the models we selected, InceptionV3 + NN performed the best. We provide the images correctly classified as defective and normal by this TL model in Figs. 5 and 7, respectively. As it can be noted, it has classified them quite well. We also provide the confusion matrix for the test results of this model in Fig. 8. As can be seen here, only six images got misclassified out of 111.

[2] https://orangedatamining.com/widget-catalog/

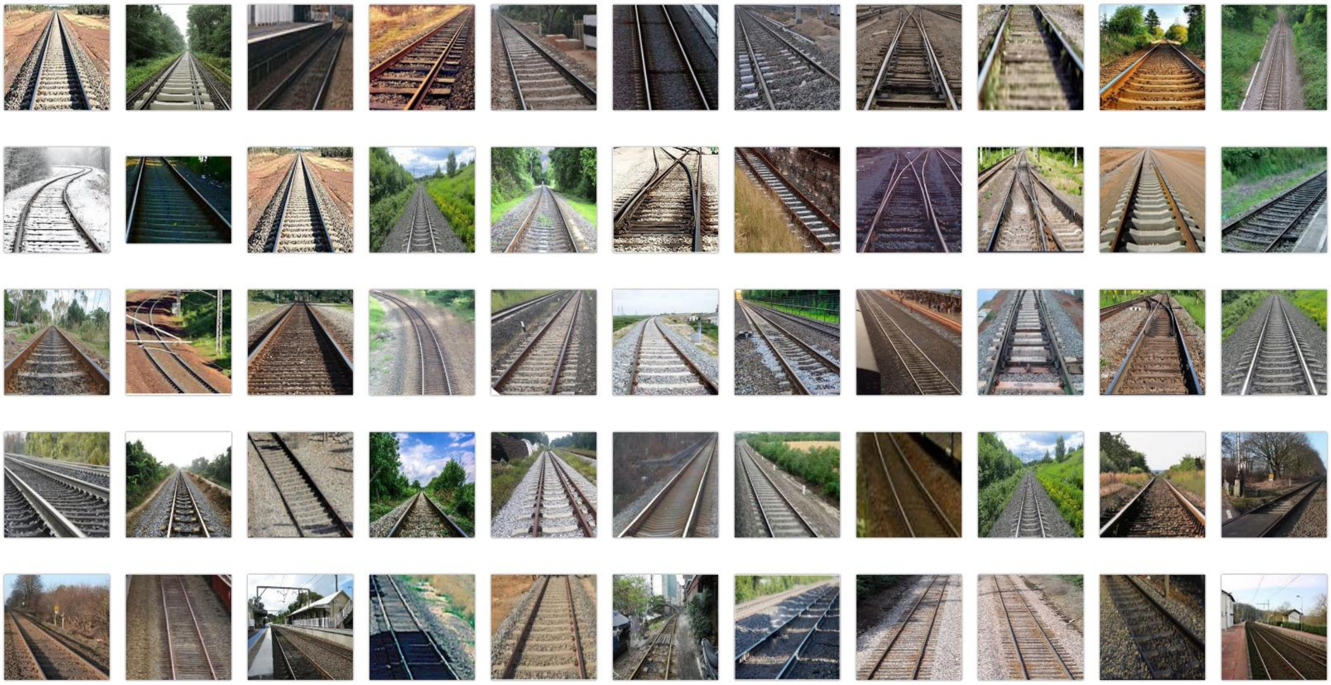
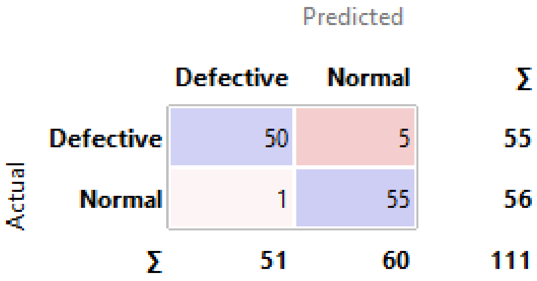


Fig. 7. Sample test images that have been correctly classified as normal by our best TL model InceptionV3+NN.



		Predicted		
		Defective	Normal	Σ
Actual	Defective	50	5	55
	Normal	1	55	56
	Σ	51	60	111

Fig. 8. Confusion Matrix of our best TL model InceptionV3+NN in the test phase. Only 6 out of 111 images got misclassified.

Out of these 6, 5 of them have been wrongly classified as normal when they were actually defective. One reason why such a thing can happen is that the network can get confused between buckling and turning tracks in some rare cases.

C. Correlation between X and Test Results

If Fig. 6, given a feature extractor, we report different correlation coefficients obtained as we vary β. Here, the correlation coefficient represents the correlation between our X scores and different test results obtained by a feature extractor for different learning algorithms. It can be seen that there are different values of β at which different feature extractors find the maximum correlation. Hence, assuming $\beta = 0.5$, i.e., considering validation accuracy alone is not necessarily the optimal approach of selecting effective TL models. It might be better to give some weightage to training accuracy too in finalizing TL models, as we did. Also, the choice of β can vary across the applications. If we don't expect much variation of images in an application such as this, β, the weight for variance error, can be on the lower side. That's another reason why we choose $\beta = 0.25$.

Conclusion

We develop a vision-based solution to detect railway track misalignments, both horizontal and vertical. The goal is to identify a couple of effective transfer learning (TL) models and then test them. First, we develop a novel dataset named TMD: *Track Misalignment Detection*[3], comprising images of defective and normal tracks. Then, we identify seven very effective models with the help of bias and variance errors modeled using training and validation accuracies. Out of these seven, the InceptionV3+NN model gave us the best test accuracy, around 94.6%.

[3] Our TMD dataset can be downloaded from the following webpage: https://sites.google.com/site/koteswarraojerripothula/neatai-servolab/track-misalignment-detection-sibgrapi-2021

ACKNOWLEDGEMENTS

This work was supported in part by the Infosys Centre for Artificial Intelligence, IIIT-Delhi, and in part by IIIT-Delhi PDA Grant.

REFERENCES

[1] K. R. Jerripothula, J. Cai, J. Lu, and J. Yuan, “Image co-skeletonization via co-segmentation,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2784–2797, 2021.

[2] K. R. Jerripothula, J. Cai, and J. Yuan, “Quality-guided fusion-based co-saliency estimation for image co-segmentation and colocalization,” *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2466–2477, Sep. 2018.

[3] D. Goyal, K. Rao Jerripothula, and A. Mittal, “Detection of gait abnormalities caused by neurological disorders,” in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, 2020, pp. 1–6.

[4] H. Rowshandel, “The development of an autonomous robotic inspection system to detect and characterise rolling contact fatigue cracks in railway track,” Ph.D. dissertation, University of Birmingham, 2014.

[5] Z. Xiong, Q. Li, Q. Mao, and Q. Zou, “A 3d laser profiling system for rail surface defect detection,” *Sensors*, vol. 17, no. 8, p. 1791, 2017.

[6] J. Ye, E. Stewart, D. Zhang, Q. Chen, and C. Roberts, “Method for automatic railway track surface defect classification and evaluation using a laser-based 3d model,” *IET Image Process.*, vol. 14, pp. 2701–2710, 2020.

[7] X. Zhang, Y. Cui, Y. Wang, M. Sun, and H. Hu, “An improved ae detection method of rail defect based on multi-level anc with vss-lms,” *Mechanical Systems and Signal Processing*, vol. 99, pp. 420–433, 2018.

[8] R. Manikandan, M. Balasubramanian, and S. Palanivel, “An efficient framework to detect cracks in rail tracks using neural network classifier,” *International Journal of Pure and Applied Mathematics*, vol. 116, no. 10, pp. 461–470, 2017.

[9] A. R. Rizvi, P. R. Khan, and S. Ahmad, “Crack detection in railway track using image processing,” *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 3, no. 4, pp. 489–496, 2017.

[10] X. Xu, Y. Lei, and F. Yang, “Railway subgrade defect automatic recognition method based on improved faster r-cnn,” *Scientific Programming*, vol. 2018, 2018.

[11] Y. Min, B. Xiao, J. Dang, B. Yue, and T. Cheng, “Real time detection system for rail surface defects based on machine vision,” *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, p. 3, 2018.

[12] L. Shang, Q. Yang, J. Wang, S. Li, and W. Lei, “Detection of rail surface defects based on cnn image recognition and classification,” in *2018 20th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2018, pp. 45–51.

[13] X. Wei, D. Wei, D. Suo, L. Jia, and Y. Li, “Multi-target defect identification for railway track line based on image processing and improved yolov3 model,” *IEEE Access*, vol. 8, pp. 61 973–61 988, 2020.

[14] Z. Lv, A. Ma, X. Chen, and S. bin Zheng, “Defect detection of pandrol track fastener based on local depth feature fusion network,” *Complex.*, vol. 2021, pp. 6 687 146:1–6 687 146:9, 2021.

[15] A. Kish and W. Mui, “Track buckling research report,” United States Department of Transportation, , 2003.

[16] “Railway investigation report r05h0013,” Transportation Safety Board of Canadan, , 2005.

[17] “Report 11/2016: Derailment near langworth,” Rail Accident Investigation Branch, Department for Transport, UK, , 2016.

[18] apheby, “Cyclic top,” https://safety.networkrail.co.uk/jargon-buster/cyclic-top/, 2016, [Online; accessed 14-Sept-2019].

[19] “Report 20/2014: Freight train derailment near gloucester,” Rail Accident Investigation Branch, Department for Transport, UK, , 2014.

[20] “Derailment of train 4vm9-vbenalla, victoria,” Australian Transport safety Bureau, , 2004.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[23] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$1mb model size,” *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360