ARTICLE TEMPLATE

Automated reading of residual plots with computer vision models

Weihao Li[a]

[a]Department of Econometrics and Business Statistics, Monash University, Clayton, VIC, Australia

ARTICLE HISTORY

Compiled January 16, 2024

ABSTRACT

TBD.

KEYWORDS

TBD

1. Introduction

The practice of plotting residuals is commonly regarded as a standard procedure in linear regression diagnostics (see Cook and Weisberg 1982; Belsley, Kuh, and Welsch 1980). This visual assessment plays a crucial role in identifying deviations from model assumptions, such as linearity, homoscedasticity, and normality. It also helps in understanding the goodness of fit and various characteristics of the model.

Generating a residual plot in most statistical software is often as straightforward as executing a line of code or clicking a button. However, accurately interpreting a residual plot can be challenging. Consider Figure 1 as an example, the residuals display a triangular shape pointing to the left. While this might suggest heteroskedasticity, it is important to avoid over-interpreting the visual pattern. In this case, the fitted model is correctly specified, and the triangular shape is actually a result of the skewed

CONTACT Weihao Li. Email: weihao.li@monash.edu

distribution of the predictors, rather than indicating a flaw in the model.

A residual plot can exhibit various visual features, but it is crucial to recognize that some may arise from the characteristics of predictors and the inherent randomness of the error, rather than indicating a violation of model assumptions (Li et al. 2023). The concept of visual inference, as proposed by Buja et al. (2009), provides an inferential framework to assess whether residual plots indeed contain visual patterns inconsistent with the model assumptions. The fundamental idea involves testing whether the actual residual plot visually differs significantly from null plots, which are created using residuals generated from the null distribution. Typically, this is accomplished through the lineup protocol. In this approach, the real residual plot is embedded within a lineup alongside several null plots. If the real residual plot can be distinguished from the lineup, it provides evidence for rejecting the null hypothesis.

The practice of delivering a residual plot as a lineup is generally regarded as a valuable approach. Beyond its application in residual diagnostics, the lineup protocol has integrated into the analysis of diverse subjects. For instance, Loy and Hofmann (2013, 2014, 2015) illustrate its applicability in diagnosing hierarchical linear models. Additionally, Widen et al. (2016) demonstrates its utility in geographical research, while Krishnan and Hofmann (2021) explores its effectiveness in forensic examinations.

However, as pointed out by Li et al. (2023), a primary limitation of the lineup protocol lies in its reliance on human judgments. Unlike conventional statistical tests that can be performed numerically and automatically in statistical software, the lineup protocol requires human evaluation of images. This characteristic makes it less suitable for large-scale applications, given the associated high labour costs and time requirements.

There is a substantial need to develop an approach that alleviates people's workload by automating repetitive tasks and providing standardized results in a controlled environment. The large-scale evaluation of lineups is impractical without the use of technology and machines.

The utilization of computers to interpret data plots has a rich history, with early efforts such as "Scagnostics" by Tukey and Tukey (1985), focusing on scatterplot diagnostics. Wilkinson, Anand, and Grossman (2005) expanded on this work, introducing graph theoretic scagnostics, which encompassed nine computable measures applied to

planar proximity graphs. These measures, including "Outlying", "Skinny", "Stringy", "Straight", "Monotonic", "Skewed", "Clumpy", and "Striated" aimed to characterize outliers, shape, density, trend, and coherence of the data. While this approach has been inspiring, there is a recognition (Buja et al. 2009) that it may not capture all the necessary visual features that differentiate actual residual plots from null plots. A more promising alternative entails enabling computers to learn the function for extracting visual features from residual plots. Essentially, this means empowering computers to discern the crucial visual features for residual diagnostics and determining the method to extract them. Modern computer vision models are well-suited for addressing this challenge.

Modern computer vision models often rely on deep neural networks with convolutional layers (Fukushima and Miyake 1982). These layers leverage hierarchical patterns in data, downsizing and transforming images by summarizing information in a small space. Numerous studies have demonstrated the efficacy of convolutional layers in addressing various vision tasks, including image recognition (Rawat and Wang 2017). Despite the widespread use of computer vision models in fields like computer-aided diagnosis (Lee and Chen 2015), pedestrian detection (Brunetti et al. 2018), and facial recognition (Emami and Suciu 2012), their application in reading data plots remains limited. While some studies have explored the use of computer vision models for tasks such as reading recurrence plots for time series regression (Ojeda, Solano, and Peramo 2020), time series classification (Chu et al. 2019; Hailesilassie 2019; Hatami, Gavet, and Debayle 2018; Zhang et al. 2020), anomaly detection (Chen, Su, and Yang 2020), and pairwise causality analysis (Singh et al. 2017), the application of reading residual plots with computer vision models represents a relatively new field of study.

In this paper, we develop computer vision models and integrate them into the residual plots diagnostics workflow, filling the gap of.... The paper is structured as follows: ...

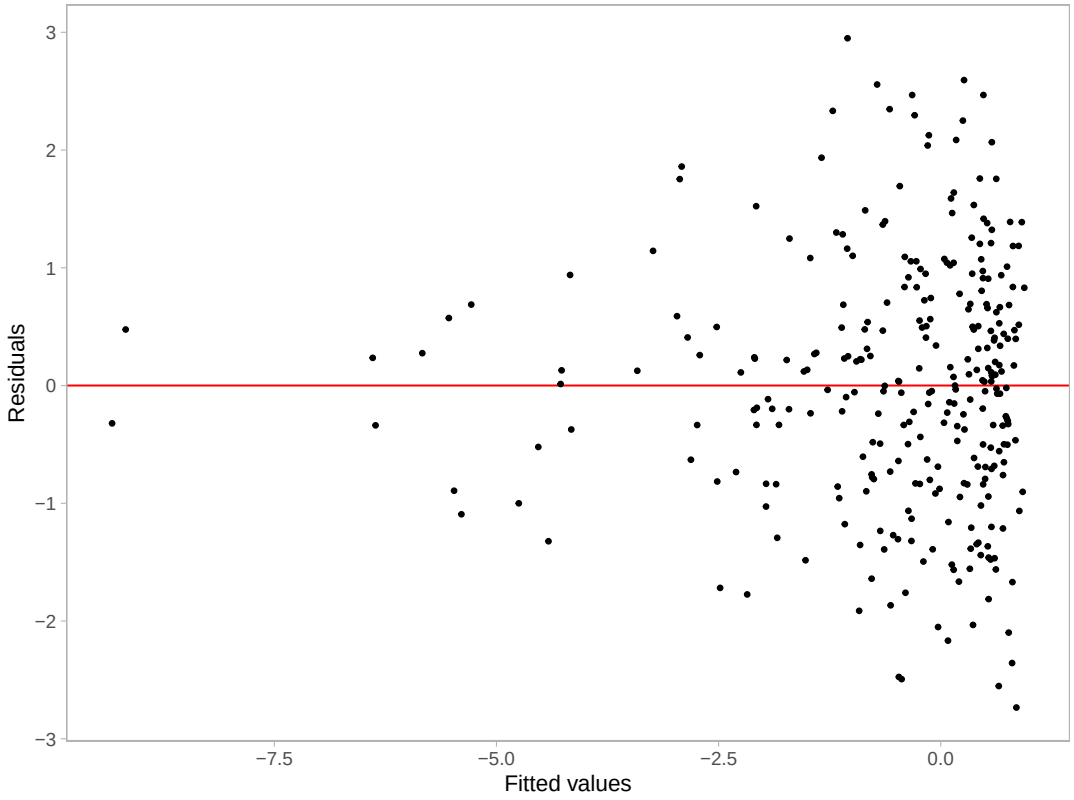

Figure 1. An example residual vs fitted values plot (red line indicates 0). The vertical spread of the data points varies with the fitted values. This often indicates the existence of heteroskedasticity.

2. Methodogology

2.1. *Distance from the good residual plots*

In a visual test, the observer will be asked to choose one or more plots that stand out as most distinct from others in a given lineup. To develop a computer vision model for evaluating residual plots within the visual inference framework, it is important to precisely define a numerical measure of "difference" or "distance" between plots. This distance can take the form of a basic statistical operation on pixels, such as the sum of square differences. Alternatively, it could involve established image similarity metrics like the Structural Similarity Index Measure (SSIM) (ref here). The challenge lies in the fact that metrics tailored for image comparison may not be suitable for evaluating data plots, where only essential plot elements require assessment (Chowdhury et al. 2018).

2.1.1. Residual distribution

The distance metrics proposed in this paper takes into account the fact that we try to measure how different a residual plot is from a good residual plot, or in other words, how different a given fitted model is from a correctly specified model. For classical normal linear regression model, residuals are derived from the fitted values $\hat{\boldsymbol{y}}$ and observed values $\boldsymbol{y}$. However, if the data generating process is known and the model is correctly specified, by the Frisch-Waugh-Lowell theorem (ref here), residuals $\boldsymbol{e}$ can be written as a linear transformation of the error $\boldsymbol{\varepsilon}$:

$$\boldsymbol{e} = \boldsymbol{R}\boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{R} = \boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ is the residual operator, $\boldsymbol{X}$ is the design matrix, $\boldsymbol{I}_n$ is a n by n identity matrix, and n is the number of observations.

One of the assumptions of the classical normal linear regression model is the error $\boldsymbol{\varepsilon}$ follows a multivariate normal distribution with zero mean and constant variance, i.e., $N(\boldsymbol{0}_n, \sigma^2\boldsymbol{I}_n)$. Using equation 1, it can be known that residuals $\boldsymbol{e}$ also follow a certain

probability distribution, which will be denoted as Q. This reference distribution Q summarises what good residuals should follow given the predictors are known and fixed.

Probability distribution Q is actually a degenerate multivariate distribution due to the fact that $rank(\boldsymbol{R}) = n - 1 < n$. This means the nth residual value will be known given the $n - 1$ residuals. To capture the exact characteristics of Q, such as moments, we can simulate a large numbers of $\boldsymbol{\varepsilon}$ and transform it to $\boldsymbol{e}$. For simplicity, we replaced $\boldsymbol{R}$ with a diagonal matrix $diag(\boldsymbol{R})$. The resulting distribution for $\boldsymbol{Q}$ is $N(\mathbf{0}_n, diag(\boldsymbol{R}\sigma^2))$.

If the model is misspecified, then the actual distribution of residuals denoted as P, will be different to Q. For example, if the model formula is correct, but the error $\boldsymbol{\varepsilon}$ follows a multivariate lognormal distribution, P will not be the same as Q and we would expect a skewed empirical distribution for residuals. And if there is an omitted variable problem, then the residual operator obtained from the fitted model will not be the same as $\boldsymbol{R}$, and again, P will be different to Q.

2.1.2. Kullback-Leibler divergence of P from Q

Define a proper distance between distributions is usually easier than define a proper distance between data plots. Given the actual residual distribution Q and the reference residual distribution P, we used a distance metric based on Kullback-Leibler divergence (ref here) to quantify the difference between two distributions

$$D = log\left(1 + KL\right), \tag{2}$$

$$KL = \int_{\mathbb{R}^n} log\frac{p(\boldsymbol{e})}{q(\boldsymbol{e})}p(\boldsymbol{e})d\boldsymbol{e}, \tag{3}$$

where $p(\boldsymbol{e})$ is the probability density function for distribution P, and $q(\boldsymbol{e})$ is the probability density function for distribution Q.

This distance metric was first proposed in Li et al. (2023). It was mainly designed for measuring the effect size of non-linearity and heteroskedasticity in a residual plot. For a linear regression model that has omitted higher-order predictors $\boldsymbol{Z}$ along

with coefficients $\boldsymbol{\beta}_z$, and an error distribution $N(\mathbf{0}_n, \boldsymbol{V})$, Q can be represented as $N(\boldsymbol{RZ\beta}_z, diag(\boldsymbol{RVR'}))$. Note that the variance-covariance matrix is replaced with its diagonal analogy to ensure it is a full rank matrix.

Since both P and Q are adjusted to be multivariate normal distributions, equation 3 can be further expanded to

$$KL = \frac{1}{2}\left(\log\frac{|\text{diag}(\boldsymbol{W})|}{|\text{diag}(\boldsymbol{R}\sigma^2)|} - n + \text{tr}(\text{diag}(\boldsymbol{W})^{-1}\text{diag}(\boldsymbol{R}\sigma^2)) + \boldsymbol{\mu}_z'(diag(\boldsymbol{W}))^{-1}\boldsymbol{\mu}_z\right), \tag{4}$$

where $\boldsymbol{\mu}_z = \boldsymbol{RZ\beta}_z$, and $\boldsymbol{W} = \boldsymbol{RVR'}$. The assumed error variance σ^2 is set to be $tr(\boldsymbol{V})/n$, which is the expectation of the estimated variance.

2.1.3. Evaluation of Kullback-Leibler divergence for non-normal P

For non-normal error $\boldsymbol{\varepsilon}$, the actual residual distribution P is unlikely to be a multivariate normal distribution. Thus, equation 4 will not be applicable to linear regression model with non-normality issue.

To evaluate the Kullback-Leibler divergence of non-normal P from Q, we need to solve equation 3. However, since $\boldsymbol{e}$ is a linear transformation of non-normal random variables, it is very common that the general form of P is unknown, meaning that we can not easily compute $p(\boldsymbol{e})$ using a well-known probability density function.

Additionally, even if $p(\boldsymbol{e})$ can be calculated for any $\boldsymbol{e} \in \mathbb{R}^n$, it will be very difficult to do numerical integration over the n dimensional space, because n could be potentially very large.

In order to evaluate equation 3 in a practically computable manner, the elements of $\boldsymbol{e}$ are assumed to be independent of each other. This assumption solves both of the issues mentioned above. First, we no longer need to integrate over n random variables. The result is now the sum of the Kullback-Leibler divergence evaluated for each individual residual thanks for the independence assumption. Second, it is not required to know the joint probability density $p(\boldsymbol{e})$ any more. Instead, the evaluation of Kullback-Leibler divergence for a single residual relies on the knowledge of $p(e_i)$,

where e_i is the ith residual for $i = 1, ..., n$. This is much easier to estimate through simulation.

The algorithm to compute equation 3 starts from simulating m sets of $\boldsymbol{\varepsilon}$ according to the error distribution. The simulated errors are stored in a matrix $\boldsymbol{A}$ with n rows and m columns. So each column of $\boldsymbol{A}$ is a set of realization values of $\boldsymbol{\varepsilon}$. Then, we can get m sets of $\boldsymbol{e}$ stored in the matrix $\boldsymbol{B}$ by applying the residual rotation operator $\boldsymbol{B} = \boldsymbol{RA}$. Furthermore, kernel density estimation (ref here) is applied on each row of B to estimate $p(e_i)$ for $i = 1, ..., n$.

Since the Kullback-Leibler divergence can be viewed as the expectation of the log-likelihood ratio between distribution P and distribution Q evaluated on distribution P, we can reuse the simulated residuals in matrix $\boldsymbol{B}$ to compute the expectation. Therefore, the Kullback-Leibler divergence can be formulated as

$$KL = \sum_{i=1}^{n} KL_i, \tag{5}$$

$$KL_i = \frac{1}{m} \sum_{j=1}^{m} log \frac{p(B_{ij})}{q(B_{ij})}, \tag{6}$$

where KL_i is the Kullback-Leibler divergence for an individual residual e_i, B_{ij} is the ith row and jth column entry of the matrix B, $q(.)$ is the normal density function with mean zero and an assumed variance estimated as $\hat{\sigma^2} = var(flatten(B))$, and $flatten(.)$ unfolds a matrix into a vector.

2.1.4. Approximation of the distance metric with a residual plot

In the previous sections, we have defined a distance metric for quantifying the difference between the actual residual distribution and an ideal reference distribution. This distance metric can only be computed when the data generating process is known. In reality, we often have no knowledge about the data generating process, otherwise we do not need to fit a linear regression model in the first place.

What we proposed in this paper is a method to approximate this distance with a residual plot. This is done by training computer vision models with a large number of

pairs of residual plots and distance. With the approximated distance, we will be able to know how different the underlying distribution of the residuals is from a good residual distribution. This is meaningful way to check if a residual plot is a good residual plot, and to know the strength of the visual signal indicating model violations.

The approximated distance is not expected to be the same as the original distance. This is largely because information embedded in a single residual plot is limited and it may not be able to summarise the characteristics of the residual distribution. For a given residual distribution P, we can generate many different residual plots. Some of them share similar visual patterns, but some of them could be visually very different from the rest, especially for models with small n. This suggests the approximated distance will vary depends on whether the observed residual plot is representative or not.

2.2. *Generation of simulated training data*

The computer vision models we developed in this study were trained with synthetic data. This means the training data and the test data were simulated from a known data generating process under different parameter settings.

We have incorporated three types of residual departures of linear regression model in this study, including non-linearity, heteroskedasticity and non-normality.

$$
\begin{aligned}
\boldsymbol{y} &= \mathbf{1}_n + \boldsymbol{x}_1 + \beta_1 \boldsymbol{x}_2 + \beta_2(\boldsymbol{z} + \beta_1 \boldsymbol{w}) + \boldsymbol{k} \odot \boldsymbol{\varepsilon}, \\
\boldsymbol{z} &= He_j(g(\boldsymbol{x}_1, 2)), \\
\boldsymbol{w} &= He_j(g(\boldsymbol{x}_2, 2)), \\
\boldsymbol{k} &= \sqrt{\mathbf{1}_n + b(2 - |a|)(\boldsymbol{x}_1 + \beta_1 \boldsymbol{x}_2 - a\mathbf{1}_n)^2}, \\
\boldsymbol{\varepsilon} &\sim N(\mathbf{0}_n, \boldsymbol{I}_n),
\end{aligned}
$$

where $\boldsymbol{y}$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{z}$, $\boldsymbol{w}$, $\boldsymbol{k}$ and $\boldsymbol{\varepsilon}$ are vectors of size n, $\mathbf{1}_n$ is a vector of ones of size n, $He_j(.)$ is the jth-order probabilist's Hermite polynomials (ref), the $\sqrt{(.)}$ and $(.)^2$ operator are element-wise operators, $\odot$ is the Hadamard product, and $g(., k)$ is a

scaling function to enforce the support of the random vector to be $[-k, k]^n$ defined as

$$g(\boldsymbol{x}, k) = 2k \cdot \frac{\boldsymbol{x} - x_{min}\mathbf{1}_n}{x_{max} - x_{min}} - k\mathbf{1}_n, \ for\ k > 0,$$

where $x_{min} = \underset{i \in \{1,\ldots,n\}}{min} x_i$, $x_{max} = \underset{i \in \{1,\ldots,n\}}{max} x_i$ and x_i is the i-th entry of $\boldsymbol{x}$.

2.3. *Different configurations of the model formula*

discuss different types of inputs and output choices, and why we choose the current design

2.4. *Architecture of the computer vision model*

2.5. *Training process and hyperparameter tuning*

2.6. *Model evaluation methods*

3. Results

3.1. *Best model performance*

- Metrics for model performance
- Shap values
- Heatmap

3.2. *Comparison with human visual inference*

3.2.1. Overview of the human subject experiment

3.2.2. Comparison

- power comparison
- decisions

3.3. *When the model works*

- simple examples (non-linearity, heteroskedasticity, . . .)
- datasaurus

3.4. *When the model does not work*

- human detect but model does not
- cartoon residuals?

3.5. *Workflow: how one use this model? (small showcase)*

4. Dicussion

5. Conclusion

- Summary of findings
- Contributions to the field
- Future directions for research

References

Belsley, David A, Edwin Kuh, and Roy E Welsch. 1980. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons.

Brunetti, Antonio, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. 2018. "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey." *Neurocomputing* 300: 17–33.

Buja, Andreas, Dianne Cook, Heike Hofmann, Michael Lawrence, Eun-Kyung Lee, Deborah F Swayne, and Hadley Wickham. 2009. "Statistical inference for exploratory data analysis and model diagnostics." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367 (1906): 4361–4383.

Chen, Yun, Shijie Su, and Hui Yang. 2020. "Convolutional neural network analysis of recurrence plots for anomaly detection." *International Journal of Bifurcation and Chaos* 30 (01): 2050002.

Chowdhury, Niladri Roy, Dianne Cook, Heike Hofmann, and Mahbubul Majumder. 2018.

"Measuring lineup difficulty by matching distance metrics with subject choices in crowd-sourced data." *Journal of Computational and Graphical Statistics* 27 (1): 132–145.

Chu, Hongyang, Xinwei Liao, Peng Dong, Zhiming Chen, Xiaoliang Zhao, and Jiandong Zou. 2019. "An automatic classification method of well testing plot based on convolutional neural network (CNN)." *Energies* 12 (15): 2846.

Cook, R Dennis, and Sanford Weisberg. 1982. *Residuals and influence in regression*. New York: Chapman and Hall.

Emami, Shervin, and Valentin Petrut Suciu. 2012. "Facial recognition using OpenCV." *Journal of Mobile, Embedded and Distributed Systems* 4 (1): 38–43.

Fukushima, Kunihiko, and Sei Miyake. 1982. "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position." *Pattern recognition* 15 (6): 455–469.

Hailesilassie, Tameru. 2019. "Financial Market Prediction Using Recurrence Plot and Convolutional Neural Network." .

Hatami, Nima, Yann Gavet, and Johan Debayle. 2018. "Classification of time-series images using deep convolutional neural networks." In *Tenth international conference on machine vision (ICMV 2017)*, Vol. 10696, 242–249. SPIE.

Krishnan, Ganesh, and Heike Hofmann. 2021. "Hierarchical Decision Ensembles-An inferential framework for uncertain Human-AI collaboration in forensic examinations." *arXiv preprint arXiv:2111.01131* .

Lee, Howard, and Yi-Ping Phoebe Chen. 2015. "Image based computer aided diagnosis system for cancer detection." *Expert Systems with Applications* 42 (12): 5356–5365.

Li, Weihao, Dianne Cook, Emi Tanaka, and Susan VanderPlas. 2023. "A Plot is Worth a Thousand Tests: Assessing Residual Diagnostics with the Lineup Protocol." *arXiv preprint arXiv:2308.05964* .

Loy, Adam, and Heike Hofmann. 2013. "Diagnostic tools for hierarchical linear models." *Wiley Interdisciplinary Reviews: Computational Statistics* 5 (1): 48–61.

Loy, Adam, and Heike Hofmann. 2014. "HLMdiag: A suite of diagnostics for hierarchical linear models in R." *Journal of Statistical Software* 56: 1–28.

Loy, Adam, and Heike Hofmann. 2015. "Are you normal? The problem of confounded residual structures in hierarchical linear models." *Journal of Computational and Graphical Statistics* 24 (4): 1191–1209.

Ojeda, Sun Arthur A, Geoffrey A Solano, and Elmer C Peramo. 2020. "Multivariate time series

imaging for short-term precipitation forecasting using convolutional neural networks." In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 33–38. IEEE.

Rawat, Waseem, and Zenghui Wang. 2017. "Deep convolutional neural networks for image classification: A comprehensive review." *Neural computation* 29 (9): 2352–2449.

Singh, Karamjit, Garima Gupta, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2017. "Deep convolutional neural networks for pairwise causality." *arXiv preprint arXiv:1701.00597* .

Tukey, John W, and Paul A Tukey. 1985. "Computer graphics and exploratory data analysis: An introduction." In *Proceedings of the sixth annual conference and exposition: computer graphics*, Vol. 85, 773–785.

Widen, Holly M, James B Elsner, Stephanie Pau, and Christopher K Uejio. 2016. "Graphical inference in geographical research." *Geographical Analysis* 48 (2): 115–131.

Wilkinson, Leland, Anushka Anand, and Robert Grossman. 2005. "Graph-theoretic scagnostics." In *Information Visualization, IEEE Symposium on*, 21–21. IEEE Computer Society.

Zhang, Ye, Yi Hou, Shilin Zhou, and Kewei Ouyang. 2020. "Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks." *Sensors* 20 (14): 3818.