XAI&I: Closing the Accuracy Gap Between Self-Explanatory AI and Black Box Convolutional Neural Networks

KOZLOWSKI, THEODOR

Supervisor: Dr Catherine Teehan

Moderator: Professor Steven Schockaert

MSc Computing

School of Computer Science and Informatics

Cardiff University

October 15, 2023

Declaration of own work

This study was completed for the MSc in Computing at Cardiff University. The work is my own. Where the work of others is used or drawn on it is attributed.

Theodor Roman West Kozlowski

Acknowledgements

I would like to thank: Dr Jacques A. Grange, Dr Mark K. Johansen, Henrijs Princis and Aissa Amadou-Dioffo for the camaraderie and fermentation for the project.

Dr Catherine Teehan for your assistance throughout.

Dr Nicholas Martin, Professor Robert Honey, Dr Qiyuan Zhang, Professor Phil Morgan, and everyone in the IROHMS and HuFEX lab for your insightful conversations and critiques.

Contents

List of Tables

1 Introduction

Innovations in Artificial Intelligence (AI) have progressed at rapid pace over the last decade, with AI systems often matching or surpassing that of human capability. The complexity of systems deployed in both public and private domains is increasing exponentially, and their use is progressively becoming interwoven within our everyday lives. Their ubiquity extends from from virtual assistants and smart speakers, to more common, yet often overlooked applications such as facial recognition, digital photo-tagging, and recommendation engines.

Despite advancements in new AI technologies, from convolutional neural networks (CNNs) to generative artificial intelligence (GAI), such as large language models (LLMs), there is an insufficient availability of easily accessible and effective tools for interpreting or explaining the decisions made by these models. This lack of interpretabilty and explainability reduces one's ability to hold an AI system and its outputs accountable, increasing the risk for misuse in fraudulent activities or the spreading of deepfakes. Furthermore, a resultant reduction in trust can lead to the potential underutilisation by experts for whom require justifications for making critical decisions, such as medical diagnosis.

As a result of the numerous concerns posed, research in the field of Explainable Artificial Intelligence (XAI) has become imperative, and in some instances, mandatory in certain domains, driven by the enaction of new legislation by governing bodies across the world ([93], [26]).

This project poses a new methodology for assessing and improving upon a specific form of XAI, that of *sequential* concept bottleneck models (CBMs). Typically, concept bottleneck models are developed through the incorporation of expert concepts into a supervised learning AI model, from which they firstly produce a set of predicted interpretable features/concepts that are subsequently used for the task of predicting an outcome/classification.

The research undertaken in this project builds upon the model proposed by Grange et al. [28], of which I was one of the co-authors. The model developed in the paper is designed with the intention of being inherently interpretable and self-explanatory, inspired and informed by a vast body of psychological research on categorisation theory, in particular that of Nosofsky et al. whose dataset [60] underpins the

work.

There were a number of limitations and questions left unanswered in the paper, such as the relationship between the concept features used, how they are interpreted and used for classification, and the effect they have upon accuracy. Incidentally one was left contemplating if, due relatively small gap in classification accuracy between the proposed self-explanatory model and that of a black-box model (i.e. one which is natively uninterpretable), if this gap could be closed through further research.

1.1 Aims and Objectives

Subsequently, the primary aim of this project was to build upon and enhance Grange et al.'s self-explanatory XAI model [28], with the intention of improving classification accuracy to match or surpass that of a black-box model, whilst preserving the capacity to predict expert-intelligible features. This endeavour aimed to further deepen upon the mutual understanding of human and AI feature abstractions.

To meet this aim, the first objective involved the development of an additional network classifier layer, of which could provide some degree of flexibility for the network to continue learning. Once completed, the evaluation of classification accuracy and concept alignment could be explored through the analysis of results using three primary research methods: manipulation of network training variables, removal of a single weakly aligned feature value from the training data, and alteration of a single feature/concept to binary or continuous/scalar in the training data.

Meeting these requirements, necessitates the development of tools to expedite data analysis, such as automation and data visualisation, enabling a deeper understanding of the effects of the proposed research methods.

2 Background

2.1 The Need for Explainable AI

To define the problem that needs to be address, a thorough literature review has been undertaken to fully expose and understand the capabilities and limitations facing the relatively new field of XAI. This requires a broad approach, considering insights from multiple academic fields beyond that of computer science, incorporating findings from the fields of psychology, sociology, philosophy, law, and politics. These investigations ultimately underpin the reasoning behind this project:

1. The need for explainable AI:
   (a) Dampen alarm amongst all sectors and demographics of digitalised societies
   (b) To fulfil existing and proposed legislative requirements
   (c) A bulwark to societal risks through the reparation of trust and enabling of accountability
   (d) To enable the productive and positive use of AI for all; acknowledging domain context, end users needs and cognitive ability
2. How should one define explainability and interpretability? The ubiquitous interchangeable use of terminology by researchers and practitioners e.g. transparency, interpretability, and comprehensibility, often used interchangeably
3. The importance of interdisciplinary collaboration and empirical evidence. This requirement needs to be fulfilled to its fullest to demonstrate the validity of the XAI research and/or development e.g. Example through the rigorous use of psychological analytical methods. I comment that one such field ripe for interdisciplinary knowledge sharing, is field of categorisation theory, amongst others such as human factors/ergonomics. XAI algorithm researchers and developers must consider the ways in which the human brain works, with many of the tasks proposed to be solved by AI previously have been done by humans, and if not solved, then used as an assistive tool, of which humans need to be able to interpret.

4. A critical analysis of the most relevant, prominently used and cutting edge XAI methods. This dissection of the research attempts to inform the design of this research project, with a pronounced focus on that of "Concept Bottleneck Networks" (CBMs) and "Convolutional Neural Networks" (CNNs). The intended goal with these methods is to accurately expose inherently interpretable image features/concepts that can be used as by humans to aid with comprehending a models decision/category classification. A broad analysis is undertaken, looking at two broad strands of methodology inherently interpretable and post-hoc models: the problems pertain to solve, their sufficiency in providing the desired level explainability to their intended users, how the underlying algorithms work and the limitations they face: by design, dataset, lack of domain knowledge, or empirical evidence garnered through experimentation.

The desire for explainability (transparency, understandability, comprehensibility, interpretability etc.) in automation is not a new field, however with the increased opacity of AI decision making and their use by all walks of life (knowingly or not), new methods are required to be able to hold to account, trust and ultimately adopt these new technologies. The term Explainable AI (XAI) was only officially defined in 2017 in a report for the United States of America's "Defence Advanced Research Projects Agency" (DARPA) [31]:

"XAI will create a suite of machine learning techniques that enables human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners".

2.1.1 Dampening alarm

The voracious pace at which Artificial Intelligence (AI) technologies continue to be developed, has raised alarm amongst experts, academics, and society at large, with little recourse to hold its resultant outputs to account. Examples of this include "Correctional Offender Management Profiling for Alternative Sanctions" (COMPAS) system used in the USA to predict a criminal defendant's likelihood of being repeat offender, of which those in the criminal justice system have become increasingly dependent on. Data analysis as shown that COMPAS makes multiple errors in classification, with an overt inclination to racially profile and mis-classify black defendants as being a higher risk of becoming recidivists compared to their white counterparts [46]. The COMPAS algorithm lacks in delivering a clear explanation

for its classification, and concurrently due to its secrecy, lacks any form of transparency, removing the opportunity for scrutiny and a sense of procedural fairness for both the defendant and the public [69]. The societal use and trust in the technology is convoluted; polarising many due to a lack of an inclusive and transparent sharing of knowledge and information, with many individuals relying solely on information extrapolated from their AI curated media "bubble", television channel or newspaper, often misinterpreting or overstating the truth for the purpose of supporting their own goals - ones attention, clicks (likes, sharing etc.), advertising or supporting a political agenda.

The development AI tools, thus far has been lead by "practitioners, disproportionately white, male, and advantaged" [47], creating a perception akin to that of a mono-culture. This lack of ideological and demographic diversity is particularly worrying if harms to society are to be effectively mitigated.

Most recent concerns observed globally have been surrounding the increased application of Generative AI in an ever-expanding set of domains. Deepfakes have arisen in a variety of contexts, from images and videos of actions that have never taken place, to the cloning of voices ([17], [82]) for the criminal manipulation of others, resulting in the divulgence of personal or classified information, often with the intent to obtain money ([23], [64]). Despite instances such as these clearly supporting a negative perception of voice cloning AI in the public psyche, there are also tremendous benefits. Take for example, a minor instance of a radio or podcast interviewee, who, with consent, will not have to return to the studio to re-record a mispronunciation or incorrect statistic, and in turn potentially saves considerable carbon emissions as a result of travel, as well as a host of peoples time and effort.

At the extreme end of this spectrum, South Korean president elect *Yoon Suk-Yeol* used a deepfake "AI avatar" in 2022, purportedly helping him to win the younger vote [80]. It's quite clear to see how this has use of AI has the potential for misuse, misleading people as to the ability or intent of a politician or influenced by nefarious actors out of sight. One such occurrence happened in June 2022 when the mayors of many European capitals were duped in a video conference by a deep fake avatar of Kyiv's, Vitali Klitschko [64]. Other domains include that which copy the style of human artists both visually and audibly e.g. fake cover versions of songs using So-VITS-SVC [79], synthesising new music in the style of artist with OpenAI's Jukebox [63], the recreation of damaged artworks [27], or mimicking of an artist's style to create new works using as thought they were alive today using OpenAI's DALL-E 2 ([14], [22]). Perhaps the most prevalent usage has been that of using ChatGPT [10] and other such Large Language Models (LLMs), to generate fake text content for the web or to write assessed work for

academics and students, despite the clear limits currently present e.g. the propensity to hallucinate false academic references ([2], [89]). The extent to which LLMs should be allowed to be incorporated is hotly debated between both students and academics [6]. Despite generative AI not being the subject focus of this research project, the rise in deepfakes has placed an even greater emphasis on the inadequacy and necessity for XAI across all domains, as but one of the many buttresses required to deliver this new technology in a responsible and accountable manner.

2.1.2 Legislation

As AI models increase in their capacity for purportedly accurate decision making, they are often so inherently complex, often referred to as the "black box problem", that even experts in the field are challenged to comprehend them. This inherent opacity has elevated the demand for greater transparency and comprehensibility due to concerns over increased over reliance, lack of accountability and potential inherent biases within the data used to train AI models. Examples that bring this into light include the European Union's (EU) General Data Protection Regulations (GDPR) that took effect in 2018, effectively giving a user the "right to explanation" [26], with proposed revision and changes discussed in the recent "Study on the impact of artificial intelligence on the infringement and enforcement of copyright and designs" by the European Union Intellectual Property Office [21]. Moves in this direction could also be seen back in 2016 when Bulgaria paved the way in requiring most government software to be open source [12], shortly followed by the creation of a New York Task force to investigate ways in which automated systems can be made more transparent [92]. For without the means from which one can be allowed to the opportunity to trust AI in an intelligible, explainable format, the result will leave the user isolated, eroding trust and hampering adoption and the merits AI can bring to society, particularly in those of a critical nature, such as health and safety.

2.1.3 Societal risks, reparation of trust and enabling of accountability

I hypothesise that one could extrapolate from research in the domains of both computer science, cognitive psychology, and governing body reports (national and global), that without a legally enforced systems for explanatory AI at all levels, there is a great societal risk of political extremism and unrest due to an erosion in trust, a lack of accountability, and subsequently the human propensity to feel a loss of control over ones surroundings ([86], [87]). The spread of fake media, and subsequently a propensity

to be vulnerable to belief in conspiracy theories. This was made quite apparent with the abundance of fake news propagated through social media, forums, and alternative news channels during the COVID19 pandemic [4]. Zellers et al.'s paper "Defending Against Neural Fake News" [58], could not have been timelier, with it's publication in 2019, just prior to the COVID19 pandemic. One of the most interesting points noted in the paper was that human readers were often prone to perceiving machine generated text as more trustworthy than human-written. Many means of automated fact checking have emerged and is well summarised by Guo et al. [32], who notes that not only text processing, but multi-modal (image, audio, and video) forms of fact checking are required, however its limitations will be dependent on large-scale annotated datasets paired with evidence beyond that of metadata being required. Van Prooijen et al. ([86], [87]) conducted research into this domain in multiple research papers. In 2015 Van Prooijen et al. conducted two studies in an applied setting, one measuring participants looking into the beliefs about conspiracy theories and public policy, and the other using a data set collected just prior to the year 2000, surveying attitudes to the threat of the Y2K bug, also known as the millennium bug, alongside beliefs in conspiracy theories that were prevalent at the time. The research shows that the "human need for control is closely coupled with their tendency to believe in conspiracy theories". In 2017 Van Prooijen et al. went further by exploring the question of why education attainment is often used to predict belief in conspiracy theories. The results in the study conclude that there are at least two mediators in this link; that of "cognitive complexity" (analytical thinking vs belief in simple solutions) and "feeling of control" (can citizens influence government, can citizens express their thoughts and feelings about government decisions etc.).

This analysis, despite being beyond the scope of XAI intended for use by experts, for which this is subject of this dissertation, remains pertinent. For if individuals and/or society are to trust and hold to account expert users of AI, an elucidation, and comprehension of the constraints and limitations of a systems functionality is imperative. This further underlines the need for a positive and transparent dialectic between governments, technology enterprises and the broader society, to disseminate accurate information as to the true capabilities and limitations of AI [93].

2.1.4 AI for all: domain context, user needs and cognitive ability

A wide variety of users and stakeholders engage with AI tools e.g. virtual assistants, smart speakers and social robots, however many fail to notice their prevalence in broader domains such as facial recognition,

digital photo-tagging, and recommendation engines [100]. As such, holding there systems accountable requires a host of different means, depending upon the domain, the user, their needs and cognitive ability ([26],[3], [35], [38], [40], [57], [81], [91], [101], [16], [33], [7]) . In a study comprised of cognitive interviews with senior and mid-career professions [36], all of whom have experience in AL and/or autonomous systems, and hold post-graduate degrees. Despite his insights being illuminating, they pertain to a limited range of user demographics (18 participants – 16 male, 2 female), showing that an explanation (global or local) is not always needed depending on the role of the individual, their style and circumstance. Trust is a key issue, particularly in the individuals developing the AI tool, with training and troubleshooting being key. However, all those questioned did desire the knowledge that they needed a "satisfactory understating of something, either the AI or the data that was fed to it, at least some of the time. The study underlines the importance of a Human-Computer Interdependence approach to development, whereby the design and evaluation of an AI (and XAI) system, is evaluated in context, including edge cases. The underlying message in this paper is that "individuals prefer to engage in explanation, rather than being passive recipients of explanatory materials" [36].

Hoffman's research however ignores the prevalence of AI outside of the professional domain. The target audience with which XAI is required is far more diverse, and should be considered the key grounding point from which one should start by first fully understanding, prior to the development of an XAI tool: who is the target audience, and what why do they need it? [7]. This requires knowledge of the user's cognitive skills, goals, and subsequent ability to understand and comprehend what the XAI tool is showing to them.

2.2 How should one define explainability and interpretability?

The challenges facing the field of XAI concern many researchers, as demonstrable with the veritable increase in papers that undertake the task of surveying and taxonomizing the wide variety of XAI approaches and definitions. Samek et al. [72] outline this very point, stating that a "theory of explainable AI, with a formal and universally agreed definition of what explanations are, is lacking".

One interpretation of "Interpretability" can be described as a passive characteristic by which a human observer can make sense of a model, a concept that can also be expressed by the term "transparency". Conversely, the term "explainability" should be understood as an active characteristic of a model, clarifying it's intent as an interface between the user and the decision making process. That is, it endeavours

to deliver an accurate proxy of the decision in a format comprehensible to humans [30].

Barredo Arrieta et al. [7] further contribute to clarification by addressing the multitude of terms frequently used in the field. "Understandability" (or "intelligibility"), should be understood as the ability for one to grasp how a model works, without the need for a full explanation of its structure or algorithmic methods for processing data. "Comprehensibility" (or "interpretability") on the other hand, is akin to the ability of an ML model's ability to present learned knowledge in a format that is inherently interpretable by humans: incorporating both quantitative and qualitative information in an integrated manner. The term "transparency" should be considered primarily when judging the varying degrees of understandabilty (or "intelligibility"): simulatable models, decomposable models and algorithmically transparent models [51].

Mohseni et al. [57] propose that common terminology can be broken down under two high-level concepts, "Intelligible Systems" and "Transparent AI" [57], both subsequently comprised of a set of desired properties and desired outcomes (see Table 1 [57]). "Interpretable AI", here defined as being a low complexity ML algorithm, and "Explainable AI", defined in a manner not dissimilar to Guidotti et al., are both categorised as practical approaches for implementing "Transparent AI" [57].

2.3 The importance of interdisciplinary collaboration and empirical evidence

It is imperative that that interdisciplinary collaboration and empirical evidence for the validity of XAI developments takes place, and should not be understated.

As AI systems increase to be seamlessly interwoven into our everyday lives, the volume of high quality peer reviewed research required in order to keep abreast of new advancements, both in bare-bones AI (CNNs, LLMs, GAI etc.) and XAI, should be undertaken in a rapacious manner. However, the promises XAI tools make of delivering meaningful and comprehensible explanations are often limited, often due to the singularity of disciplinary insight of the authors. To counter this issue, experts from a diverse range of disciplines should be called upon, from computer science, machine learning (ML), human-computer interactions (HCI), psychology, philosophy, sociology, law, politics and so on [9]. Diverse perspectives are essential in giving insight on the multifaceted challenges that each academic, and professional domain faces, as well as society as a whole [47].

Two disciplines that are progressively engaging in collaborative endeavours are, computer science, encompassing areas such as ML and HCI, and psychology ([28],[35]). These combined efforts are driving the development of some of the most advanced XAI tools, with the aim of incorporating psychological theories of human cognition, such as categorisation, decision making, and measurements of trust, alongside novel AI algorithms and mixed modalities. Experimental-psychologists are in a unique position, possessing the requisite tools to empirically test XAI systems in experimental settings. Furthermore, ethical domains such as philosophy and sociology are increasingly entering the discourse ([47], [48], [8], [104]), providing a crucially needed evaluation of AI legitimacy and safety within the domains it is employed [29].

Various methods and tools for psychometric analysis and evaluations of AI and XAI have recently been developed, particularly in regards to the concept of trust ([38], [43], [35], [65]). The concept of "trust" is intriguing in its ambiguity; a multidimensional factor of on going debate as to its meaning, particularly in the field of Human Factors and Ergonomics. Few AI researchers undertake a considered approach when assessing trust in their work, with many referring to a single definition by Lee and See [49] [85].

The most commonly used tool in AI research has been the "Trust between People and Automation" scale (TPA) [41], however a more recent metric has been developed with the specific purpose of evaluating XAI, the "Trust Scale for Explainable AI" scale (TXAI) [38]. Despite TPA being the most commonly used, the majority of AI studies rely on defining their own questionnaires and definitions of trust, making it almost impossible to validate one AI model against another [85]. In light of this, TPA and TXAI have been evaluated empirically in the context of AI for the first time through an online study comprised of 1368 participants [65]. Results from the study suggest that TXAI (an 8 item questionnaire, using a 5 point Likert-type scale) is most fit for purpose, however the 6th item ("I am wary of the AI") should be removed, leaving no negatively worded items in the scale, preventing misinterpretation by participants and miscoding by researchers alike. These suggestions are congruent with the set of best practices proposed by Schrum et al. [105], who's research into the use of Likert scales for statistical analysis of human-robot interaction (HRI) experiments, is highly transferable to the domain AI.

		Two-Factor EFA			Single-Factor EFA	
No.	Item	PA1	PA2	h2	PA1	h2
1	I am confident in the AI. I feel that it works well.	.84		.73	.86	.73
2	**The outputs of the AI are very predictable.**	.45	-.36	.27	.36	.13
3	The AI is very reliable. I can count on it to be correct all the time.	.80		.62	.78	.61
4	I feel safe that when I rely on the AI I will get the right answers.	.85		.72	.85	.72
5	**The AI is efficient in that it works very quickly.**	.52		.28	.53	.28
6	**I am wary of the AI.** (R)	.33	.46	.38	.42	.17
7	**The AI can perform the task better than a novice human user.**	.69		.47	.69	.47
8	I like using the AI for decision making.	.81		.70	.84	.70

Note: Problematic items are marked in bold. PA1/PA2 = loadings on the first/second factor; h2 = communality. Reverse-coded items are marked with (R).

Figure 2.1: 8 item scale from Hoffman et al [38]. As presented in: [65].

Complementary to the assessment of trust, Klein et al. [43] have set out a framework of 11 requirements by which to guide researchers to run smaller efficient experiments to evaluate human-AI work systems. This new methodology of "Minimum Necessary Rigor" for empirical evaluation of AI is considered necessary in light of the limitations of existing practices which suffer from at the consequence of "rigor mortis": requiring significant funds, large participant pools and complex design and often providing answers no longer required by the time of completion. This method is somewhat contrary to common practice, which promotes the use of a calculating a required sample size for achieving a statistically significant effect on a parametric test [19]. This new methodological approach should be being welcomed in light of the voracious pace of AI development, however there is yet any evidence in the literature of its use in practice at time of writing. As such, one should see this proposal as a springboard for discussion, enticing academics to put it into practice and potentially embark on further well scoped, and replicable evaluation methods. Therefore until adequate evidence of its sufficiency is obtained, one should be sceptical until proven otherwise.

In order to build out human centred XAI systems numerous studies and frameworks have been built ([57], [85],[37], [38], [13]) with strong evidence suggesting that feedback modalities with AI systems are key to the their use, with the ability to converse "improving comprehension, acceptance trust and collaboration" [101]. However the use of LLMs as the sole tool for human computer interaction with AI poses significant ethical risks in light of the current lack of XAI methods being deployed [102].

The fundamental purpose of XAI tools, beyond that of academic research, is that they are intended for practical use in real world domains, in high-stakes decision making, such as medical diagnosis, automated driving, detecting deep-fakes or LLM generated text. This fundamentally requires incorporating the user

from inception to deployment.

High-stakes decision tasks are a prime use case for XAI, yet due to their nature, often problematic to study due to the requirement of developing experiments that are abstracted from the reality.

Examples such as ChatCAD [91] demonstrate the possible capabilities in the realm of medical diagnosis and engagement, enabling patients whom may not have access to a doctor the ability to question and garner further insight into their condition. However, ChatCAD has not been discussed nor reviewed by medical practitioners, highlighting the need for far greater interdisciplinary collaboration.

Leichtmann et al. conducted an exploratory study into the use of XAI [50], assessing the use of XAI for the high-stake task of deciding if a mushroom is edible or poisonous. The XAI interface delivered an attribution-based example using Grad-CAM [74], of which the results were statistically significant in improving participants performance in mushroom picking. There are however multiple limitations with this study, such as being restricted to conducting it online rather in the field, and by the sole use of Grad-CAM, of which despite performing better than LIME [68], still pertains the intrinsic issues that befall all visual mapping tools.

Alternative XAI methods such as LIME, SHAP, SmoothGrad and Integrated Gradients that are more frequently used have garnered a larger pool of critical research, of which is tackled in the following chapter.

2.4 Prior research - the foundation of this project

To understand the motivations and subsequent approach that this project took, it is necessary for one to grasp the concepts, reasoning, and methodology of the research paper on which it builds upon, along with the limitations therein.

In 2022 Grange et al. [28] published a paper presenting a novel methodological approach for constructing an inherently understandable AI. Motivated by the desire to improve AI accountability, increase trust, and adoption, primarily in the high-stakes decision making applications involving health, safety and risk.

The research approach was primarily grounded in the field of Psychology, drawing upon a body of research conducted by Nosofsky et al. ([73], [62], [56]) whose expertise and publications in the realm of human categorisation theory had recently highlighted the potential interest of Deep Neural Networks (DNNs) to the field.

Human categorisation theory encompasses various mathematical models, such as mixed representations and rules of the mind([5], [20]), as well as prototypes [78] and exemplars ([59], [55], [45]), of which the latter two are grounded upon the notion that categorisation is based upon similarity judgements. Prototype and exemplar model domain research has predominantly been constrained to the use of artificially created categories for systematic assessment and review due to the nature of the complexity of real-world categories.

Recent advances in DNNs have been employed to predict human similarity judgements [73], revealing the possibility of shared underlying properties for similarity judgements between AI systems and humans. This subsequently prompted psychologists to further explore the depths of these mutual interpretations. Grange et al. demonstrated this explicitly through a partially feature constrained model, which exhibited a strong correlation with the 8 feature dimensions identified by Nosofsky et al. [56] and the resultant multi-dimensional scaling (MDS) co-ordinates [73].

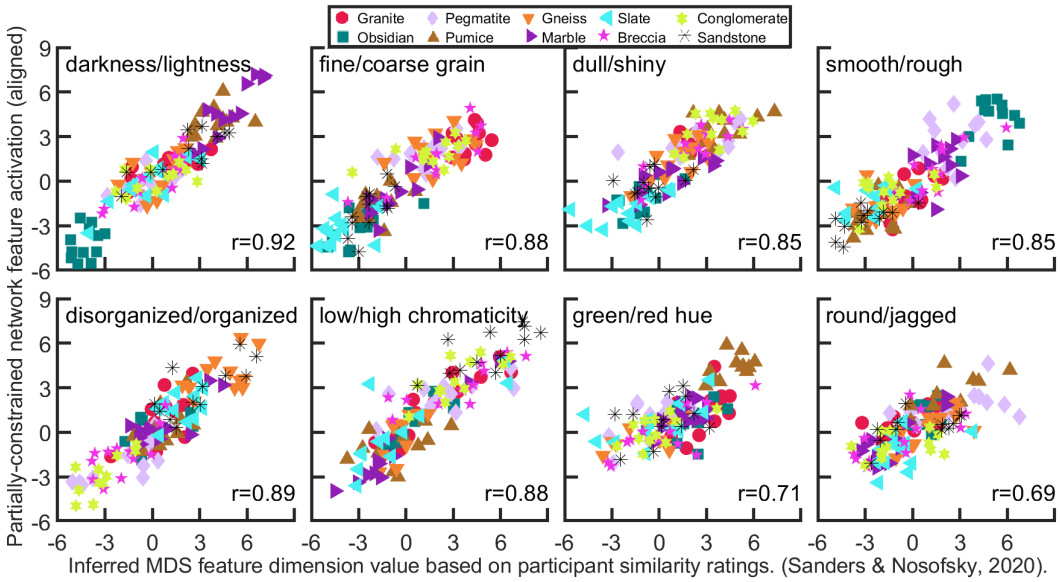


Figure 2.2: Abstracted features are found to be an affine transform of the Sanders and Nosofsky (2020) inferred MDS feature dimension values based on naïve participants' similarity ratings. Each plot specifies the MDS feature dimension and the Pearson's r correlation coefficient. As presented in: [28].

A prerequisite of any explainable and intelligible AI system is the establishment of a shared understanding of concepts. Humans, in general, are less adept at articulating their motivations for categorisation due to complex reasoning, lacking a formal basis on which to do so. On the contrary, domain experts are often intrinsically more adept to do so, equipped with knowledge needed to establish a basis for more profound

comprehension of shared concepts between AI systems and experts.

Grange et al. chose the domain of rock classification primarily due to the availability of Nosofsky et al.'s real-world labelled dataset of images and concept/feature ratings, with assessed similarity properties. The similarity judgements from naïve participants enabled the collation of a large database of MDS co-ordinates, from which to input to a low-dimensional similarity space. Due to the high cost of building such a database, Nosofsky et al. circumvented the need to acquire further similarity ratings for new category instances by developing a DNN that successfully predicted MDS co-ordinates [73].

It's worthy of note that the presence of expert-identified features, from which naïve participants made similarity judgements, expedited participants natural category learning [56]. As such, further motivating the use of this domain and dataset from which to develop an inherently understandable AI.

Despite the impressive quality of the ratings data, the collection of rock images was limited in scale when compared to those used to train other AI models e.g. CUB [88], which comprises of n = 11,788 bird photographs. Nosofsky et al.'s set of rock images used in this paper significantly restricted, with 160 "full size" per rock category (n = 1,600), of which 320 random patches were generated and subsequently augmented (translated, flipped, rotated and scaled) to create a set of 480 (n = 4,800).

Thorough employing transfer learning, the penultimate layer of 2048 node activation's (referred to as the "Average Pooling Layer") were utilised as input into a neural network. These activation's, denoted as "X train", were combined with a dataset of manually annotated expert feature concepts (Y train), to train the network to predict equivalent expert features. These constrained predicted-feature concepts were subsequently utilised as training data for a single-layer classifier. The fully feature constrained network performed well, achieving a classification accuracy mean of 85.1% (SD 0.7%), only 1.7% short of an unconstrained network's accuracy mean of 86.8% (SD 0.8%).

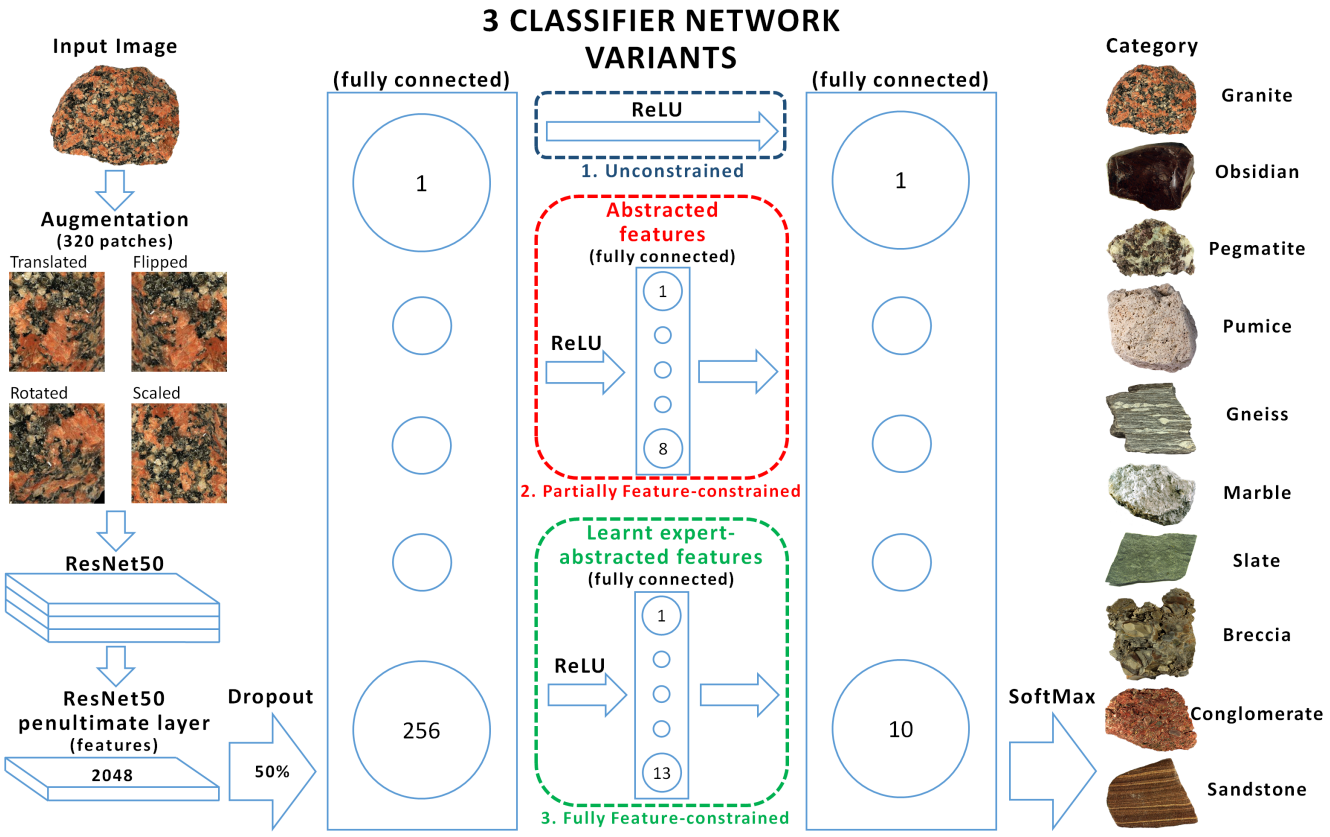


Figure 2.3: Illustration of the three classifier network variants making use of transfer learning with Renset50. Rock images are first augmented before being fed through Resnet50, the penultimate layer of Resnet50 is taken through a 50% drop-out layer, a fully connected 256-node layer, then one of three paths before a 10-node category activation layer is passed through a SoftMax function: (1) an unconstrained variant path, where the 256 node layer fully connects to the 10-node layer through a ReLU function, (2) a partially feature-constrained variant path, where the 256-node layer activations are condensed, via a ReLU function, down to an 8-node, 'Abstracted Features' layer and (3) a fully feature-constrained variant path, where the abstracted feature layer is forced to be made of transfer-learned expert-abstracted features. As presented in: [28].

The finding that an accuracy mean accuracy gap of only 1.7% between that of the constrained and unconstrained black-box model is impressive. There is however a lack of analysis of features, assessing if any may overlap in knowledge representation, particularly when the correlation between the expert feature ratings and the predicted feature ratings is below that of those which highly correlate e.g. "Brightness Heterogeneity" (r=0.43) compared to "Roughness" (r=0.92). Further review is also required to understand the nature of features that are rated in a continuous scalar fashion (soft coded) i.e. "Average Grainsize", compared to that of those with binary ratings (hard coded) e.g. "Presence of Crystals". A considerable body of research in similar methods shall be critiqued in this dissertation, in support of both binary and scalar methods of coding concepts.

This begs the question of whether DNNs are interpreting non-binary concept/feature ratings beyond binary distinctions, as many of the features in this model lean towards continuous ratings. However, one should consider if the DNN is perceiving features that are visible in the image or features that are intrinsic to the category/classification type. For example, “Granite” always contains crystals, yet the extent to which they are present may not be equally visible from every image angle.

2.5 A Review of Relevant and Recent XAI Methods

As XAI developers and researchers alike strive to develop new insight and tools with the aim of delivering improvements on the interpretability and explainability of models for a wide variety of users, there ultimately lies a distinction between which route they choose to take to deliver on these promises. This varies dramatically based upon the domain scope (local or global), stage (ante-hoc or post-hoc) and the output format (numerical, visual, textual or mixed) [96].

Chen at al. [11] propose defining interpretability and explainability methods based upon stage alone:

1. Ante-hoc, inherently interpretable models ([11], [44], [34], [70], [18], [28])
2. Post-hoc explanations for existing neural networks ([68], [52],[99],[76], [77], [74])

The majority of XAI methods fall into the category of post-hoc explanations, with vastly different levels of explanation given with equally as wide levels of comprehensibility. Post hoc methods, such as heat maps and saliency maps ([68], ,[99],[76], [24],[77], [74], [84]), being the most common and often used within industry [39].

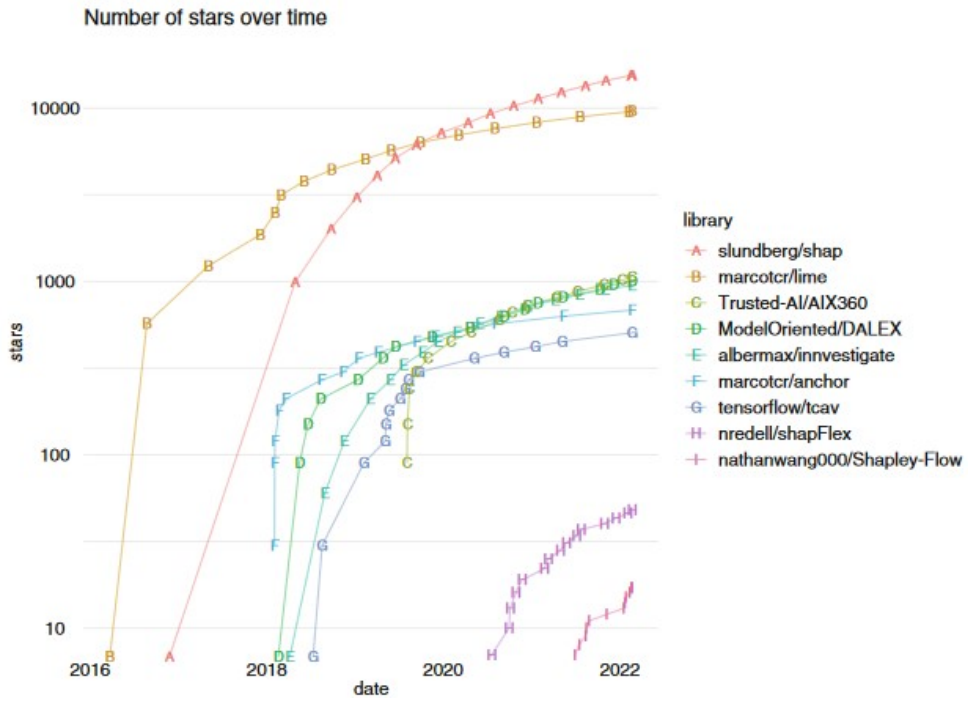


Figure 2.4: The most popular XAI repositories on GitHub (number of stars) as presented in: [39].

This methodological approach tackles the delivery of an explanation by weighting each pixel of an input image, producing a visual image of to the end user, with the aim of showing the importance each pixel made to the final classification. This can be problematic, as many of these methods can only highlight the edges of an image, offering little of use in terms of an explanation i.e. the differences between one class and another ([71]. Saliency maps are also prone to to sensitivities in the data or model, and as such, potentially misleading if used as a sole means of assessment [1].

The misleading nature of saliency maps is supported by Kim et al. [42] in an experiment measuring the perceived importance of an image and concept. Participants were shown four images with concept captions (produced using Concept Activation Vectors [42]), each with two corresponding saliency maps (SmoothGrad [77] and Integrated Gradients [1]) and asked to rate the importance of the image to the model (10-point Likert scale), the importance of the caption (10-point Likert scale), along with their confidence in their answer (5-point Likert scale). The results indicate that due to the percentage of correct answers rated as very confident being equivalent that of incorrect answers, that saliency maps have a tendency to be misleading. There was also a degree of decorrelation in accuracy between the two saliency maps i.e. when one correctly communicated a concept, it was not always true with that of the

other. This further supports the notion that post-hoc saliency methods are frequently fragile and limited in their ability to deliver a useful, explanation. This supports the need for inherently interpretable models, of which this research project attempts to tackle.

2.5.1 Concept Bottleneck Models

The emergence and growth in research on Concept Bottleneck Models (CBMs) has been motivated with two key benefits in mind when approaching the task of XAI. Firstly, there is the promise that class classification predictions can be explained using high-level human interpretable concepts produced by a concept predictor [28]. Secondly, that a human operator can intervene, altering concept values and monitoring the effect this has on the final prediction [34].

One of the first propositions in this field was that of “Concept Whitening”(CW) [11]. This purportedly inherently interpretable model, places a bottleneck in a CNN, replacing a batch normalisation (BN) layer with a novel CW layer. By doing so, the latent space of a neural network is “disentangled”, constraining the neurons in the network to realign to the axes of predefined concepts understandable by humans. The research method uses ResNet18, with experiments conducted that assert that the 16th layer provides the clearest approximation of concept alignment. This focus on the penultimate layers of the model, is congruent with the idea that higher-level semantic concepts are present in the latter layers of CNNs ([103], [28]). All experiments using this combination are trained using the Places 365 dataset, with either three or seven simultaneous concepts from the MS COCO dataset.

It can be argued that the concepts being defined here are quite primitive and limited in dimensionality e.g. “air-plane”, “bed” or “person”, and as such fail to deliver, beyond that of the clarity of a single concept in an image, and is somewhat analogous to a "grandmother cell" in neuroscience. It therefore falls short of denoting the deeper semantic descriptors of an images features e.g. if the image is that of an air-plane, why is it so? Presence of wings, flying in the sky, shape etc. Rather the output focuses on what it is not e.g. an air-plane is not a table, bed or boat. This is a clear limitation of the training data used (Places 365 & MS COCO), of which is such granular details are not inclusive. Consequently, the paper concludes, that if a full explanation on each computation were to be delivered then this would lead to a restriction on flexibility, however one may determine that the complexity would be considerably reduced if only applied within an a single expert domain. The transference of the CW model to an expert domain is attempted, focusing on skin lesions, however it is limited by the use of only two concepts,

age and legion size, of which age is discovered to play no importance to classification. Opposingly, the size of the legion (>=10) is in the third quartile of importance (of 512 axes), a concept that is know to be used by physicians for diagnosis of skin lesions [90]. This brief insight into the importance of expert domain knowledge reinforces the need for the incorporation of well defined concepts based upon existing empirical evidence, reinforcing the body research undertaken in this project.

The pursuit of a delivering a comprehensive explanation has led to the prolific development of new concept bottleneck models (CBM), both ante and post-hoc ([44], [54], [34], [98]).

Koh et al. [44] offer three CBM models for exploration: (1) Independent bottlenecks, whereby concepts and classification are learnt in separate algorithms, with the resultant learnt concepts used for classification at test time. (2) Sequential bottlenecks, whereby the concept is learnt, with the learnt concepts used to inform classification learning. (3) Joint bottlenecks, whereby the concept and classification are both learnt in combination during training, whereby λ regularisation factor is adjusted using the combined losses

The findings illuminate that not only by using human annotated data for training, but along with the intervention by an expert (particularly independent bottlenecks), a model can be updated if an artefact is incorrectly identified as a concept/feature, substantially improving accuracy beyond that of a standard model, with an overall observation that there is not in fact a trade-off between high task accuracy and high concept accuracy [44].

Due to the improved levels of predictive performance (analysis of Root Mean Squared Error - RMSE) through concept refinement, Koh et al. take preference to the proposed joint CBM model. Yet it is noted that the benefits of intervenability in joint models is detrimental to performance. To the contrary, independent models benefited the most from expert intervention of concept prediction using ground truths.

The quality of the output of all three models proposed by Koh et al. is interrogated by Margeloiu et al.'s [54] three desiderata of a CBM: 1. Interpretability: Being able to note which concepts are important for the targets. 2. Predictability: Being able to predict the targets from the concepts alone. 3. Intervenability: Being able to replace predicted concept values with ground truth values to improve predictive performance.

The methodology used by Margeloiu et al. to critique Koh et al.'s three CBM models, requires the development of a concept oracle model (CO); a model using ground truths to predict target images. The

CO is used to compare through the correlation of the root mean square error (RMSE) of each CMB model and CO. The finding suggests that independent CMBs and COs have a far higher coefficient of determination. This observation aligns with the hypothesis that predicted concepts (as per sequential and joint CBMs) are not used as intended, but rather as proxies to incorporate target information, shown with a vastly reduced RMSE as the level of concept intervention increases, particularly that of independent CBMs (see Fig 4 in [44]). It is suggested by Margeloiu et al. that one of the reasons for low correlation of concepts in sequential and joint CBMs to COs is a result of the use of one-hot encoding for concept values i.e. concepts are binary (0 or 1). Leading on to suggestions of further study, through the analysis of the variability of concept representations by means of binary and scalar concepts, and one-hot encoded categoricals. It should be noted that some analysis of binary and scalar concepts is tackled later on within this research project manuscript. Margeloiu et al. also take on the use of saliency maps (Integrated Gradients with Gaussian Noise baseline) to try and assess the predicted concepts visually, showing attention across the whole image rather than that of the defined feature, however, as commented by many others, saliency maps are unable to map concepts in any meaningful manner ([71], [1],[42]).

2.5.2 Post-hoc Concept Bottleneck Models

Complementary to the development of CMBs, much research has been undertaken in the development of post-hoc CBM (PCBM) methods ([34], [98], [15]), of which attempt to address multiple issues, such as the loss of concept clarity or "leakage" [53]. It is speculated that the cause of leakage may be the result of an overlap of concepts in the concept predictor layer due an insufficient concept set being available e.g. some classifications may require more concepts than others, or that "soft" concepts i.e. non-binary concepts (probabilities, often between 0 and 1) allow unintended information to be conveyed. In conclusion, resultant leakage muddies both interpretability and the ability to effectively intervene on the concept predictor. The alternative to soft CBMs is that of hard CBMs, whereby concepts are binary. One may consider this approach to be more truthful, as leakage between concepts is prevented, yet this method is prone to lower classification accuracy unless all concept details are captured correctly. There is no flexibility for the system to leak knowledge into another concept, thus thorough domain knowledge is required to prevent miscoding concepts, either by not getting the quantity correct, through human error in labelling the training data, or a mutual lack of understanding between the user, the domain, the input, and the models deliverables (perhaps there is hidden knowledge the model requires that is not innately

intelligible by humans). This notion is contrary to the supposition by Margeloiu et al. that hard CBMs may be responsible for the low correlation of concepts in sequential and joint CBMs. A conclusion may be drawn that naturally, some concept features could always be considered binary due to their intrinsic nature e.g. obsidian rocks will always have a glass like texture, or marble being composed of crystals. However, it is less clear if the network would benefit from further granularity, i.e. how much of said concept is present in the image used for training.

Havasi et al. [34] propose two further means of addressing leakage and improving performance of hard concept CBMs: (1) a side channel model and (2) an auto-regressive architecture. Prior to using either tools a model is first analysed to see if the concepts are sufficient in predicting the final label classification, or if more information is desired – the lack of a Markovian assumption. The side channel model is a small single layer which is trained concurrently with the hard CBM from a set of latent concepts, with the flexibility to infer how many concepts are required for label prediction. This side-channel enables one to estimate the completeness of the original concepts, and therefore can be used as a form of diagnosis to infer if there are key concepts missing. Yet, on closer inspection it is clear that much of the side channel information has the propensity to deliver concepts uninterpretable by humans – something that may or not be desirable depending upon the domain and accuracy required by the end user. The second tool is that of an auto-regressive architecture that allows for hard CBMs to learn correlations between concepts, therefore re-weighting concepts, with interventions also affecting concept predictions of prior concepts. They note that normalisation is of upmost importance to ensure that correct predictive distribution. Both methods come at the cost of increased computation; however they do put hard CBMs at an accuracy level equivalent to that of soft CBMs. This may be desirable in some domains whereby concept knowledge is always translatable in a binary manner, however in many domains this is not always the case, and if applied then the initial mandate, to enable human interpretability, is lost.

A novel post-hoc CBM (PCBM) model is proposed by Yuksekgonul et al. [98], attempting to tackle three key limitations of CBMs: access to data i.e. the laborious task of annotating or collating a concept dataset, increasing performance, and the enablement of intervention by human input. The first task of collating a concept databank is approached in two ways, through utilising Concept Activation Vectors (CAVs) or with state of the art (SOTA) multi-modal models such as "Contrastive Language Image Pre-Training" (CLIP) [66]. The use of a CAV is utilised by learning concepts selected by a domain expert, or automatically from the data, that positively or negatively associate with the image. This method

is pragmatic in that it does not require the training data to mirror the data used to train the backbone model, as per requirement of a CBM. A linear standard vector model (SVM) is then used to learn the corresponding CAV, using the positive and negative examples to denote the vector notal to the linear classification boundary.

The second method for constructing a concept bank is through the utilisation of the text encoder from the multi-modal model CLIP, of which contains both text and image encoders to map a description to a shared embedding space. A concept description vector is obtained through collating the relevant text embeddings to be utilised in combination with an open knowledge graph, ConceptNet [83], to collate the relationships between classes and concepts.

Once a concept subspace is learnt through either CAV or CLIP, a sparse linear model or decision tree is used for prediction due to the inherent clarity and insight with which one can observe a decision being made. It is noted that the richness of the concept subspace is important to the performance accuracy of the PCBM, of which would no doubt disincentivise any potential user for uptake over that of a normal model. An attempt to solve this issue is made by utilising a sequential residual predictor, that attempts to retain some of the original model's accuracy, a concept denoted as Hybrid Post-hoc CBMs (PCBM-h). Despite this effort to debug the CBM model, there is still the risk that one may be limited with a poor concept library, ill equipped to express and solve the task it is required, and potentially containing and reinforcing biases. Despite this, Daneshjou et al. [15] successfully incorporates the PCBM methodology into a novel work flow, using expert dermatologist defined concepts to develop SkinCon. A dataset of existing image annotations was used, and clarified using expert intervention..

Others have approached the matter of concept completeness/sufficiency and discovery in alternative means, such as ConceptSHAP [97]. Yeh et al.'s method delivers a completeness score, shedding light on how sufficient concepts are for delivering an explanation, along with a means of to alight how important each concept is to an input by adapting the widely used Shapley value method [75]. The method uses concepts that are common across all classes as opposed to one, postulating that shared class concepts are useful for interpretation of the model, e.g. one may determine through analysis of the concepts by nearest neighbour, that the shape of an animal's head is important and shared between multiple classes. In an expert domain, the idea of shared concepts is often common place due to the scope being local rather than global. A quick alternative to this method, often used, is to calculate the L2 norm of a the weights in a model to deride which features are most important. However, unlike Shapley values, L2 norm cannot

capture the nuanced interaction between individual features in the same way. As such, the incorporation of ConceptSHAP into the realm of CBMs is arguably a vital addition to the AI and ML practitioners toolbox.

2.5.3 Concept Bottleneck Models and Large Language Models

SOTA CBMs have begun leveraging the use of Large Language Models (LLMs), such as aiding with the laborious task of annotating data sets. Language Guided Bottlenecks (LaBo) [95], does so by aligning GPT-3's sentence based concepts using CLIP to form a bottleneck layer. There are some limitations, such as fine tuning the output of GPT-3, restricting the sentence length, and preventing the use of class names. Yet this is arguably preferable as the generated concepts can be controlled and chosen based upon a number of factors, such as interpretabilty, classification accuracy, and those which are highly discriminative and recognisable by CLIP. As such, this model strongly leans towards the the classification of being ante-hoc/inherently interpretable, as it essentially focuses on the fine tuning of CLIP. However the performance is restricted by the training data of GPT-3 (at least in this iteration), which excels in common categories, yet rapidly falls short with delivering higher granularity in more nuanced fields of enquiry, reducing it's potential use case for a number of expert domains. The concept of integrating LLMs, despite being out of the remit of this project, is one of significant future interest, particularly with the development of new LLMs that can be fine tuned to focus on specific expert domains. Yet, as previously noted, LLMs are vastly more difficult to explain in comparison to current NN methods, requiring whole new methodologies in the realm of XAI [102].

Other developments incorporating LLMs explore the incorporation of multiple models and multiple input modalities e.g. ChatCAD [91] for medical image diagnosis. ChatCAD leverages multiple SOTA computer aided diagnosis (CAD) networks e.g. a disease classifier, lesion segmentor, and report generator, from which a combined prompt text can be generated for input into an LLM such as ChatGPT. In doing so a condensed report of the diagnosis is produced, leveraging the the models knowledge of the medical field [25], and enabling the user to interrogate the report through conversational enquiry.

Zhang et al. [101] support the notion of free-form conversation with a network, demonstrating that comprehension, acceptance, trust and collaboration are significantly improved between the user and AI model. However their experiment is not without limitations, such as the use of the wizard-of-oz methodology and only two two feature attribution tools (LIME and Grad-CAM) being used as a means of explanation

(the limitations of which have previously been stated).

Further developments in multi-modal LLMs such as NExT-GPT [94] indicate the possibility to deliver upon potentially higher levels of explainability as well as inclusivity via the use of modality switching e.g. from natural language text, to image, video or audio, subsequently delivering a more human like means of interaction. NExT-GPT does so by the use of modality-switching instruction tuning (MosIT), in combination with a manually curated high quality dataset. Despite the impressive possibilities of this model (requiring only 1% of the parameters to be updated during training for each new modality), with hopes of incorporating additional modalities in its next iteration e.g such as tables, figures, heat maps and so on; the manual curation of a bespoke dataset (MosIT) is somewhat concerning. As with all data sets, there is always the risk of human bias being encoded, unconsciously or otherwise, through the selection of data and its subsequent annotations. This further highlights the need for legislation and the regulation of data used to train such models.

With the ever increasing complexity of AI systems that combine multiple models and modalities ([91], [94],[95]), it will become an ever increasing challenge to incorporate explainability and interpretabilty throughout. Therefore, the use of accurate, explainable, and interpretable CBMs hold a strong defence to be included as an essential item in a developers toolbox when delivering a trustworthy complex AI system, offering a glimmer of transparency, in light of other elements which may appear opaque.

3 Approach, Methodology and Results

3.1 Approach

The approach to enhancing network accuracy whilst preserving human aligned concepts, was formulated through a series of informal discussions with departmental colleagues and co-authors as noted in [28]. Consequently, both I and others, were already invested in the idea, so the decision to commit to the further utilisation of the existing model that had been researched and developed made logical sense. The decision to approach the project using this existing constrained sequential CBM and expert-rated data set [28] as a starting point provided me with the advantage of being able to meet the aims and objectives of the dissertation within the given time frame, without building a network from scratch.

The use of this model and subsequent dataset also had the distinct advantage of empirical evidence for the validity of the feature concepts in the research conducted by Nosofsky et al. ([60], [56], [73])., The data set was also unique in its use of combining both *soft*, information rich, scalar concepts (in this instance between 1 and 9), and *harder* binary concepts (1 or 9). The reasoning for using a scale of 1-9 is rooted in the initial dataset of MDS co-ordinates from Nosofsky et al. It was, and remains, unclear if this a-typical method of rating feature concepts improves performance in comparison to other *soft* CBM models, of which use probability ratings between 0-1. It is also worthy of noting that in previous iterations of the network (in the build up to [28]) that a model gradients function was used, penalising any feature concepts that were rated as -1 i.e. not present. However this method of rating concepts was removed due to the analysis showing a reduction in the accuracy of classification and weak concept alignment. Evidence in the literature for improved classification accuracy when using *soft* scalar concepts has been pointed to as a potential result of information *leakage* [34] i.e. soft concepts may be misrepresenting the data, and therefore muddying the interpretability of the model. Yet with binary concepts, despite being more truthful, they often penalise the accuracy of the classification. This has been hypothesised as a potential result of the lack of required concept features needed by the model. The discovery of this in the literature only served to strengthen my initial thoughts on the matter and concluded that it was a worthy avenue of research within this project.

Amongst other proposals considered, was that of joint training through a combined loss function, akin to Koh et al. [44]. However, I believed there was a pressing need for a deeper understanding of what the existing sequential CBM model had learnt with regards to the validity of the concepts. This was further supported by claims from Margeloiu et al. [54] that joint CBMs have a tendency towards a low correlation of concept values, and that the benefits that may be had from intervention are in fact detrimental to performance.

The desire to understand more about the sequential CBM developed by Grange et al. and the concepts it had learnt, eventually fostered the novel idea that the weights and biases ($w + b$) learnt by the existing CBM [28] could be used to initialise a new CBM classifier. The decision to take this approach was based on the notion that by giving the network some additional degrees of freedom to learn, greater classification accuracy may be achieved (akin to that of a black box) whilst regaining the existing highly correlated expert feature values. Additionally, the appended hybrid network could serve as a tool for assessing the quality of the learnt expert feature values in the sequential bottleneck model via comparative correlation. In turn, allowing the user to intervene with the training data.

The framework of the approach adopted comprised of the following steps:

- Utilisation of the model proposed by Grange at al. 2022 [28] as a foundational starting point
- The development of an additional network classifier that employs the acquired knowledge embedded within the weights and biases of the previous CBM model [28]. Implementation in MATLAB [106], following the methodology of the previous model, will expedite development and subsequent data collection
- Developing analytical tools with which to appraise the classification accuracy and concept alignment of the new hybrid classifier to that of the prior CBM and to the expert training data
- Quantifying the effects on classification accuracy and concept alignment when using binary or continuous concept/feature ratings
- Evaluating the impact on classification accuracy when removing a concept/feature with low correlation to the input dataset of human intelligible features
- Investigating the outcomes on classification accuracy and concept alignment through the manipulation of network learning variables, such as the number of epochs, mini-batch size, and learning

rate

Throughout the project an ongoing literature review was conducted in order to keep abreast of new developments in the field of XAI. The lack of clarity with regards to terminology in the domain was a considerable hindrance during the development of the initial sequential CBM [28]. As such it was essential that I made a continued effort to build a repository throughout the project to inform my decisions and prevent the replication of research already conducted in previous studies. I found many examples of post-hoc CBM methods ([34], [98], [15]), and others that addressed the loss of concept clarity or "leakage" [53], however I did not find evidence of the methodology as proposed in this work, being replicated elsewhere.

3.2 Design and Implementation

The development of the hybrid classifier network was completed in Matlab as planned. A snippet of the code can be see below:

```
%% Unconstrained hybrid with preset weights
function net = trainUnconstrainedHybrid(categoriesToUse,totalTrain,
    networkExpertPrediciton256Weights, networkExpertPrediciton256Bias,
    networkExpertPrediciton13Weights, networkExpertPrediciton13Bias,
    netC2Weights, netC2Bias, networkExpertPredictions)
disp("Training hybrid network...");
 if ~exist('nrExpertFeatures','var')
       nrExpertFeatures = size(networkExpertPredictions,1);
  end

layers = [featureInputLayer(2048),
    dropoutLayer()
    fullyConnectedLayer(256),
    reluLayer(),
    fullyConnectedLayer(nrExpertFeatures),
    fullyConnectedLayer(10),
    softmaxLayer,
    classificationLayer()];

% The 10 categories (using 9 out of the 12 original images), labels repeated 9
    times per category = 90 labels
```

```
lbls = repelem(categoriesToUse,9);

% Adding the 320 augmented image labels (from 1-9, as per previous full sized
    images) = 28820 + 90 = 28890
lbls = [lbls,repelem(categoriesToUse,320*9)];

% As per above, appending the additonal set of Nosofsky image labels to the
    array = 40 + 28890 = 28930
lbls = [lbls,repelem(categoriesToUse,4)];

% Then add the augmented images of the additional set = 12800 + 28930 = 41730
lbls = [lbls,repelem(categoriesToUse,320*4)];

options = trainingOptions('adam', ...
    'MaxEpochs',50,...
    'MiniBatchSize', 1024,...
    'InitialLearnRate',10^-3,...
    'Verbose',false,...
    'Plots','training-progress');

layers(3).Weights = networkExpertPrediciton256Weights;
layers(3).Bias = networkExpertPrediciton256Bias;
layers(5).Weights = networkExpertPrediciton13Weights;
layers(5).Bias = networkExpertPrediciton13Bias;
layers(6).Weights = netC2Weights;
layers(6).Bias = netC2Bias;

layers(3).WeightLearnRateFactor = 0;
layers(3).BiasLearnRateFactor = 0;

net = trainNetwork(totalTrain',categorical(lbls),layers,options);
end
```

Listing 3.1: Matlab code for the Hybrid Network Classifier

The design developed continues the use of transfer learning as per [28], taking in the 2048 features from ResNet50, with a 50% dropout layer, in line with the previous CBM design. However, where the new

hybrid classifier differs is in that the three fully connected layers are initialised with the weights and biases ($w + b$) learned from the constrained sequential CBM (see Figure 3.1). The first fully connected layer consists of 256 nodes which are not allowed to learn (with a learning rate of $\eta = 0$). In contrast, the subsequent fully connected layers, of 13 nodes in the concept layer and 10 nodes in the classification layer, are free to do so (with an initial learning rate of $\eta = 0.001$). This flexibility in the model to deviate along only two of the feature layers (concept and classification layers), using the same number of nodes as per the previous CBM (except for the case of one experimental method of removing one concept, and subsequently one node in the concept layer), made it possible to evaluate the results against both the original CBM and the expert training data. This design ensured that the resultant concept features were prevented from straying too far outside of the bound of 1-9, as per the original training data.

One may view the hybrid classifier as an extension of a sequential CBM, due to the manner in which it relies upon the weights and biases of the preceding network classifier, thus in reference to the terminology used in the literature one could pertain that this model functions as a post-hoc sequential CBM.

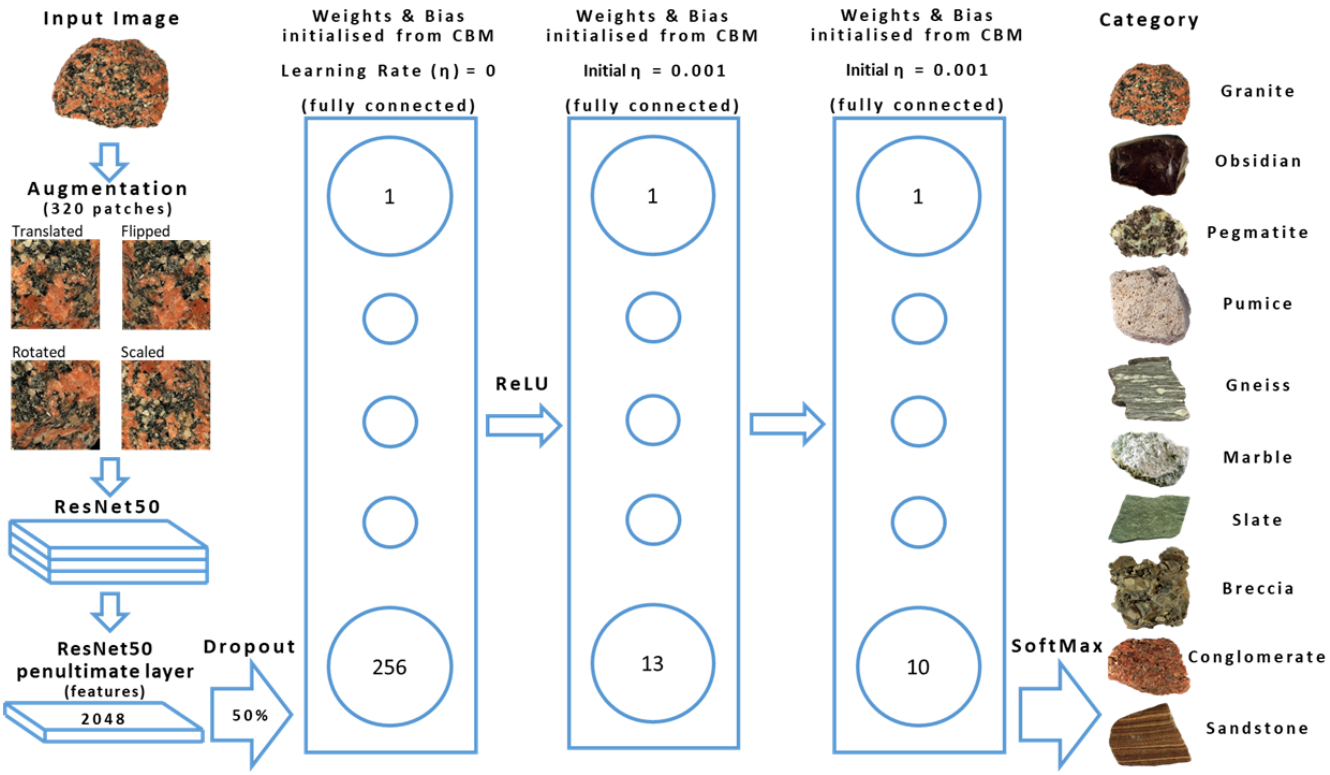


Figure 3.1: The framework of the hybrid CBM classifier

Concurrently an partially feature constrained black-box model was developed akin to that in [28] (see

figure 2.3). The number of feature nodes was set to 13 in order to enable a a fair comparison to that of the hybrid CBM.

For one to achieve a sufficient set of data to equate to fair analysis of variance, the code was automated (see code 7.2) to complete 12 runs of 12 alternating validation sets. The output of each of which required collation and manipulation for analysis of accuracy, and concept alignment between each of the networks and that of the original set of expert ratings (an example of the training data used for Granite can be seen her 7.1)

Analysis of the immense collation of data from each permutation and manipulation of the models required the development of a plethora of Jupyter notebooks insights it held(7.1, 7.2, 7.3), with which manipulate the data and visualise it using the Python library *Plotly* [107] (see Appendix subsection: 7.2). This served as in order to expedite insight at a far greater pace.

3.3 Results

My initial investigation took the form of manipulating training variables such as the number of epochs, mini-batch size and learning rate.

Through the analysis of training data visualisations it was clear that validation accuracy of the original CBM was diverging from the training accuracy as I increased the number of epochs, a sign of the over-fitting the data. As such, the correlation of the output features with that of expert features was not negatively effected, with insignificant variance. The opposite was true when reducing the number of epochs, with a sweet spot between 175-200. As such, I decided to stick to the existing use of 200 epochs, as this made it simpler when comparing to previous iterations.

The manipulation of learning rate was of little benefit to the existing CBM model, reducing performance when the number of epochs was increased, yet was not significant enough to warrant further exploration. I believe this in part to be the benefit of the *Adam Optimiser* [108], of which upon research appears to be a common effect due to its inherent ability to self-tune. Therefore I concluded that leave the learning rate at 10^{-3} (0.001) was adequate.

The manipulation of mini-batch size is another method typically used to prevent validation accuracy from diverging from training accuracy. I explored the use of reducing the size down to 256, which appeared to decrease divergence, however was of no benefit to performance. Upon further reading, many note that in some instance some over-fitting can be beneficial to a model, and as I was going to explore the benefits of appending the hybrid classifier, I returned to using a mini-batch size of 1024.

For further analysis, some of the results of these manipulations can be seen in the appendix 7.3.2.

Despite there being no grand finding from the manipulation of training variables, the results from the use of the hybrid network were however successful, with the aim of achieving an improved accuracy in classification that is on par to that of a black-box model. The 13 feature hybrid CBM classifier achieved a classification accuracy of 87.8% (SD 0.58%), an improvement of 1.27% in comparison to the partially feature-constrained model of 86.53% (SD 4.39%) and unconstrained network's performance (mean 86.8%, SD 0.8%). Interestingly the 12 feature network, with the weakly correlated n the concept of *Brightness Heterogeneity* removed (Continuous crystal rating CBM network $r = 0.43$, Hybrid CBM network $r = 0.3$, Binary crystal rating CBM network $r = 0.45$, Hybrid CBM network $r = 0.27$, 7.2) performed

well achieving a higher classification accuracy mean of 88.36% yet with a larger standard deviation of the means of 3.41%.

	13 Features		Unconstrained 13	12 Features	
	Continuous	**Binary Crystal**		**Continuous**	**Binary Crystal**
Runs/Val Sets	12/12	12/12	12/12	12/12	12/12
C2					
Epochs	200	200	200	200	200
Learning Rate	0.001	0.001	0.001	0.001	0.001
Learning Rate	10^-3	10^-3	10^-3	10^-3	10^-3
Minibatch Size	1024	1024	1024	1024	1024
Accuracy Mean - Val Set	83.66%	84.63%	**87.15%**	82.25%	82.48%
SEM - Val Set	0.52%	0.53%	0.83%	0.52%	0.57%
Accuracy Mean - Run Set	83.66%	84.63%	**86.53%**	82.25%	82.48%
STD of Means - Run Set	0.57%	**4.03%**	**4.39%**	**3.57%**	2.79%
SEM - Run Set	0.16%	1.16%	1.27%	1.03%	0.80%
Hybrid					
Epochs	50	50		50	50
Learning Rate	0.001	0.001		0.001	0.001
Learning Rate	10^-3	10^-3		10^-3	10^-3
Minibatch Size	1024	1024		1024	1024
Accuracy Mean - Val Set	87.80%	87.59%		87.55%	**88.36%**
SEM - Val Set	0.68%	0.70%		0.65%	**0.62%**
Accuracy Mean - Run Set	**87.80%**	87.59%		87.55%	**88.36%**
STD of Means - Run Set	0.58%	**2.94%**		3.43%	**3.41%**
SEM - Run Set	0.17%	0.85%		0.99%	0.98%

Table 3.1: Classification accuracy ratings comparing network variables. The first two columns use 13 expert feature ratings for training and compare implications of using continuous or binary ratings for the presence of crystals. The "13 Unconstrained" column represents a black-box model, limited only by the constraint of using a layer of 13 nodes prior to classification. The last two columns use 12 expert feature ratings, with the most weakly correlated feature "Brightness Heterogeneity" removed. A comparison is also drawn as to the implications of using continuous or binary ratings for the presence of crystals.

4 Analysis

To gain further insight as to what was happening with regards to the validity of the concepts I undertook some analysis of the importance of the weights, looking at L2 regularisation. The results below indicate towards a clearer lack of complexity for the network to understand the importance of feature concepts i.e. due to the hybrid network having lower weights (closer to 0) so prevents some over fitting is prevented perhaps. Further granular analysis of this scan be viewed in the appendix (7.3.7.

	sCBM	Hybrid
All Concepts	**Sum L2**	**Sum L2**
Granite	6.0152	5.9691
Obsidian	2.5668	1.5327
Pegmatite	14.0439	10.8745
Pumice	12.7726	8.2170
Gneiss	12.2029	9.6233
Marble	19.6444	15.0057
Slate	18.1869	12.0342
Breccia	9.0980	7.0099
Conglomerate	7.7457	6.1521
Sandstone	11.0264	8.7511
Total L2	**113.3028**	**85.1696**

Table 4.1: Analysing the complexity of concepts between the sequential CBM and Hybrid Classifier CBM using summed L2 values for all concepts. Data from a single run of a validation set (Images 1,2,3)

5 Conclusions

Upon completing my research and experimentation within the realm of self-explanatory AI, it has become self-evident that employing tools like concept bottleneck models have the potential to greatly contributes to the fostering trust in AI through the clarity and correlation of which they can convey human interpretable data. However, the use of the variety of XAI tools is without doubt very domain dependent. A key example of this is the critical domain of medical diagnosis and use of imaging for disease detection. As it stands the vast majority of AI tools used within the healthcare diagnosis domain come in the form of saliency maps [39]. I believe that to truly foster trust and subsequently facilitate the use of AI for informed and improved decision making , multiple explanatory tools will be required, and tailored to the relevant stakeholders, be they medical doctors, lawmakers, data scientists or the public. The addition of explanatory data such as feature concept ratings from a model trained on “human in the loop”/expert data (self-explanatory, concept bottleneck etc.), the accuracy the model gives for its classification, as well as saliency maps (individually or collectively) to reinforce what the model has detected.

Yet one should be careful as to not overload the individual with too much information., as this may hinder decision making; say for example if the models don’t correlate with each other in diagnosis (feature concepts don’t correlate with saliency maps), or at the other end of the spectrum, the accuracy of the AI is so high that trust has bequeathed to an over reliance on AI tools resulting in misdiagnosis. The combination of methods is perhaps one way in which to combat over reliance, as referencing will always require a human in the loop, and as previously alluded to, if models do not correlate multiple experts may be required to communicate as the their interpretation with individual expert knowledge.

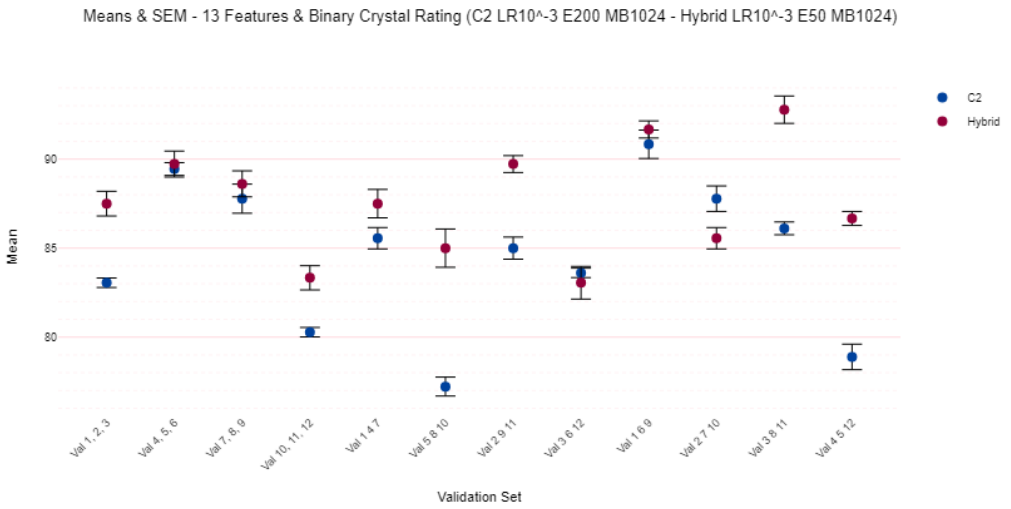
To include a broader range of categorisations (in the domain of rocks), one would presume that additional nodes and concepts may have to added to distinguish the finite differences between rocks of similar appearance visually. The challenge of a humans ability to distinguishing one rock type from another was highlighted by Nosofsky [61] using MDS co-ordinates,

Despite the improvements in performance by the hybrid classifier network, there were still some instances where by the sequential CBM outperformed, possibly due to the quality of the image or the ratings data itself: "all dataset-based methods are limited by the diversity of examples in the dataset used and

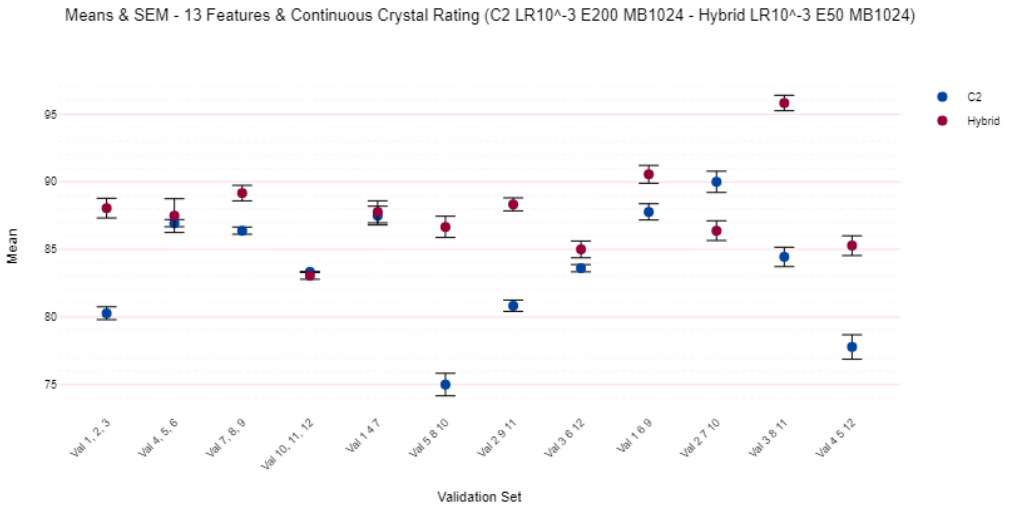
the quality of labels." [67]. The unpicking of this unfortunately evaded my capabilities in the time available.

Instances where C2 outperformed Hybrid

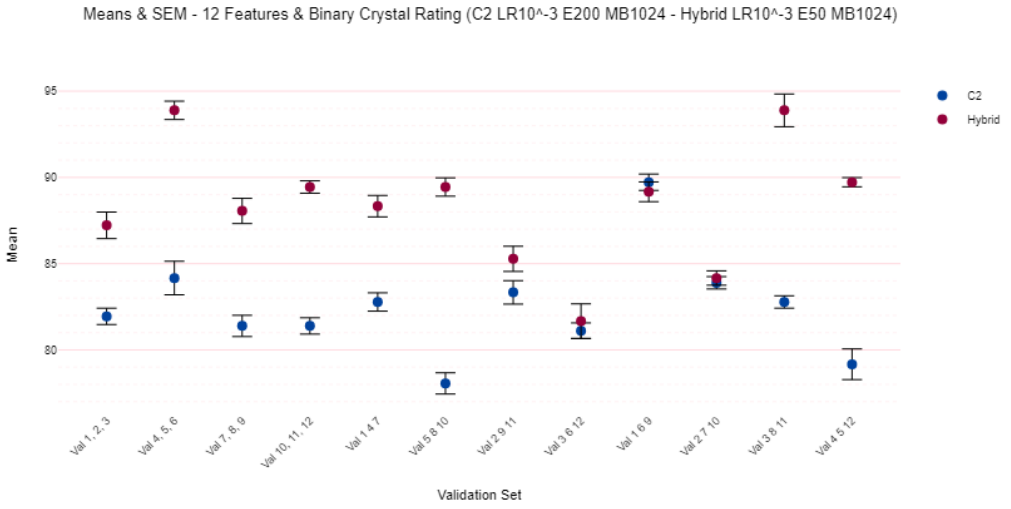
- 13 Features & Binary Crystal Rating
  - Val 3, 6, 12 = C2 (mean = 83.61%, SEM = 0.27%), Hybrid (mean = 83.1%, SEM = 0.92%)
  - Val 2, 7, 10 = C2 (mean = 87.78%, SEM = 0.72%), Hybrid (mean = 85.56%, SEM = 0.60%)



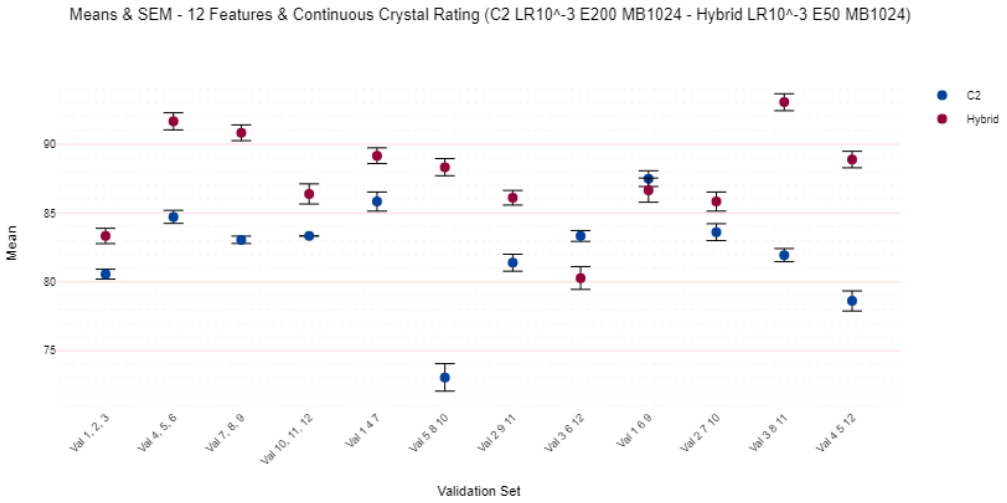
- 13 Features & Continuous Crystal Rating
  - Val 2, 7, 10 = C2 (mean = 90.0%, SEM = 0.79%), Hybrid (mean = 86.39%, SEM = 0.73%)



- 12 Features & Binary Crystal Rating
  - Val 1, 6, 9 = C2 (mean = 89.72%, SEM = 0.47%), Hybrid (mean = 89.17%, SEM = 0.57%)



- 12 Features & Continuous Crystal Rating
  - Val 3, 6, 12 = C2 (mean = 83.33%, SEM = 0.39%), Hybrid (mean = 80.28%, SEM = 0.83%)
  - Val 1, 6, 9 = C2 (mean = 87.50%, SEM = 0.57%), Hybrid (mean = 86.67%, SEM = 0.88%)



In hindsight I would have have attempted to incorporate ConceptShAP [97], or another method for assessment of concept evaluation.

The increased use case for LLMs is of great intrigue, particularly with the development of multi-modal models, however this will require a far deeper dive by XAI researchers if the outputs are to be worth of trust.

6 Reflection and Learning

At the outset, the undertaking of this project was quite daunting, partially due to its inherent complexity, tackling a topic far beyond the technical scope of any of the taught modules covered in the MSc. I was therefore acutely aware that I was to be challenging myself considerably, both technically and academically, requiring me to learn at pace to deliver on my research aims.

The process of delivering this project has given me valuable insight into the breadth of skills required to deliver a program of research. It very quickly became apparent that time management and organisation skills were essential, along with the ability to be agile with ones aims and objectives.

The use of a Gantt chart was helpful in enabling a broad overview of progress. However, I quickly discovered that when embarking on a novel research area that is inherently complexed and challenging to both one's technical skills and domain knowledge, the intricacies of the tasks can require altering priorities at short notice. For example, running the required volume of iterations of the network could take anywhere between 2-3 days depending on how many permutations and training variables were altered. If a bug appeared in the the code, particularly one that I was not aware of due to the use of a remote machine, my workflow would have to pivot to allowing time for debugging. This made planning unwieldy at times, as each collation of results required data analysis from which I used to inform on the next permutations of the networks.

In an ideal world the data analysis tools and skills to monitor the CNNs output would be a prerequisite before commencing with the research. However, the bespoke data analysis tools required for the task were not available from the outset, as I was not yet equipped with the skills and agility required to build them. As such, it initially took substantial time to complete analysis, using a combination of excel and Jupyter Notebooks, before fully transitioning to the latter. This limitation in my skill set served as a vital learning curve, as the desire to complete analysis at pace, drove me to learn a host of new skills in data manipulation. The acquisition of these new analysis skills, particularly in the visual format, served to reinforce my knowledge of data visualisation acquired during the taught phase of the MSc. The subsequent understanding from these visual aids made me vastly more agile, enabling me to hypothesise on the implications of my network alterations, and to re-aligning my objectives to fit within the limited

time frame of the project.

Initially the ambiguity of the time frames required for the network to run and the data to be analysed often meant that I found myself neglecting the Gantt chart and using a combination of handwritten notes and task lists, word documents comprised of hypotheses, and technical summaries, windows notepads for reminders of which permutation of the model was running on the remote machine, alongside a host of annotations in the code. As the whole process of data collection, analysis, literature acquisition and review became more fluid, I found myself returning to the Gantt chart as means of keeping on top of progress. In hindsight, I vastly overestimated my ability to complete the scale of research that I desired in such a short time frame. The discovery midway through the project that the manipulation of training options was not necessarily the best means to achieve increased accuracy, at least in this instance, redefined the scope of the project. Subsequently I made the executive decision to pivot my attention to analysing the effect of removing a weakly correlated feature concept and of altering concepts ratings as binary or scalar. Prior knowledge of neural network optimisation methods before undertaking this project would have saved a great deal of time and effort, however I learned a great deal in the process. I am now considerably more informed and skilled as making alterations to the inner working of CNNs and AI systems in general, of which I hope to apply in my work beyond this project.

At the start of the project I was only briefly acquainted with Matlab, having used it for assisting co-workers in research activities. The subsequent development of the project in Matlab gave me a far greater understanding of the language and environment for the undertaking of research and development work. Despite this, it did hinder my ability to experiment with new methodologies and tools discovered in my literature search, of which the majority were typically developed in Python (and freely available on GitHub). Given the time and the opportunity I would preferred to have re-written the project in Python, and to have incorporated a means of using Shapley values [97] to assess the model's dependence on a concept/feature.

Due to my involvement in the conference paper [28] that inspired this dissertation, I was put in good stead to further embark on a wider literature review, to inform the necessity and relevancy of my work within the academic landscape. My thirst for knowledge and enthusiasm for discovering new methods and ideas in XAI put my time management skills to the test when undertaking the acquisition of literature for review. Upon reflection, I believe that my compulsion to understand the broader narrative as well as the granular, was essential, as it became apparent from reading the vast platitude of papers published on

AI and XAI, that few in the field fully consider the inherent interdisciplinary nature of their work and the broader landscape in which it is sited e.g. psychology and philosophy.

The discovery of papers on covering self-explanatory or interpretable XAI models and methods that were conceptually relevant was a complexed task. The terminology used by authors is frequently used interchangeably, with multiple interpretations of their meanings thus posing a great challenge when making comparative evaluations of models. As such I decided to include a subsection within this dissertation addressing the issue, which not only aided my understanding, but hopefully the readers too.

The lack of commonly used methodologies by which to assess XAI approaches empirically was also preventative in assessing XAI models on any common ground. The field of Human Factors, HCI and Robotics have many tools at their disposal with which to use for empirical research, however those in XAI often appended their own methods for assessing the credibility of their work. I sincerely hope that this will soon change with the very recent publication of papers addressing the issue ([38], [85], [65], [43]).

As my literature review was ongoing throughout the project, with my task of developing the hybrid sequential network already set out at the beginning of the project, it was near impossible to back track and incorporate any of the new tools that I subsequently discovered in the literature. The discovery of Koh et al.'s [44] paper was somewhat disheartening, with the proposal of a sequential CBM vastly akin to that proposed by Grange et al. [28]. However the lack of grounding in the psychological domain i.e. the research conducted by Nosofsky et al. set's the papers apart significantly. The discovery Koh et al.'s paper did however inform me of the term "Concept Bottleneck Network", of which greatly assisted my literature search due its adoption by subsequent researchers.

As such, the project has up-skilled me immensely, with an insight and knowledge to build my own tools and tackle debates over the preference of one methodology over another.

Throughout the project I lacked the benefit of garnering any insight for external experts (except at its inception) or empirical evidence with which to evaluate the validity of my models output e.g. conferring with geologists over the concepts used or measuring the skills of geologists to correctly classify the rocks by image alone (typically done by a variety of physical methods rather than visual alone).

lessons about the topics addressed in the project where not already covered by the substance of your dissertation (underpinning theory or philosophy; value of approaches; understanding gained; problems

not solved; effectiveness; etc.).

7 Appendix

7.1 Code Examples

7.1.1 Matlab code for the Hybrid Network Classifier

```
%% Unconstrained hybrid with preset weights
function net = trainUnconstrainedHybrid(categoriesToUse,totalTrain,
    networkExpertPrediciton256Weights, networkExpertPrediciton256Bias,
    networkExpertPrediciton13Weights, networkExpertPrediciton13Bias,
    netC2Weights, netC2Bias, networkExpertPredictions)
disp("Training hybrid network...");
 if ~exist('nrExpertFeatures','var')
      nrExpertFeatures = size(networkExpertPredictions,1);
  end

layers = [featureInputLayer(2048),
    dropoutLayer()
    fullyConnectedLayer(256),
    reluLayer(),
    fullyConnectedLayer(nrExpertFeatures),
    fullyConnectedLayer(10),
    softmaxLayer,
    classificationLayer()];

% The 10 categories (using 9 out of the 12 original images), labels repeated 9
    times per category = 90 labels
lbls = repelem(categoriesToUse,9);

% Adding the 320 augmented image labels (from 1-9, as per previous full sized
    images) = 28820 + 90 = 28890
lbls = [lbls,repelem(categoriesToUse,320*9)];

% As per above, appending the additonal set of Nosofsky image labels to the
```

```
   array = 40 + 28890 = 28930
lbls = [lbls,repelem(categoriesToUse,4)];

% Then add the augmented images of the additional set = 12800 + 28930 = 41730
lbls = [lbls,repelem(categoriesToUse,320*4)];

options = trainingOptions('adam', ...
    'MaxEpochs',50,...
    'MiniBatchSize', 1024,...
    'InitialLearnRate',10^-3,...
    'Verbose',false,...
    'Plots','training-progress');

layers(3).Weights = networkExpertPrediciton256Weights;
layers(3).Bias = networkExpertPrediciton256Bias;
layers(5).Weights = networkExpertPrediciton13Weights;
layers(5).Bias = networkExpertPrediciton13Bias;
layers(6).Weights = netC2Weights;
layers(6).Bias = netC2Bias;

layers(3).WeightLearnRateFactor = 0;
layers(3).BiasLearnRateFactor = 0;

net = trainNetwork(totalTrain',categorical(lbls),layers,options);
end
```

Listing 7.1: Matlab code for the Hybrid Network Classifier

7.1.2 Automating 12 runs of 12 sets of validation images

```
cd D:\Theo\XAInI\Scripts\AllNetworkTraining
max_Runs = 12; %max_Runs = 1;
runs = 1;
set = 1;

for x=1:max_Runs
    validation_Sets = {[1,2,3] [4,5,6] [7,8,9] [10,11,12] [1,4,7] [5,8,10]
   [2,9,11] [3,6,12] [1,6,9] [2,7,10] [3,8,11] [4,5,12]};
```

```
    for i=validation_Sets
        training_Tokens = [1,2,3,4,5,6,7,8,9,10,11,12];
        validation_Tokens = i{:};
        for j=validation_Tokens
            if ismember(j, training_Tokens)
                training_Tokens(training_Tokens == j) = [];
            end
        end
        disp("Set " + set);
        disp("Run number " + runs + " of " + max_Runs);
        disp("Using validation image numbers " + strjoin(string(
    validation_Tokens)));
        cd D:\Theo\XAInI\Scripts\AllNetworkTraining
        networksAll
        clearvars -except runs max_Runs set;
        runs = runs + 1;
        if runs > max_Runs
            runs = 1;
            set = set+1;
        else
        end
    end
    if set > max_Runs
        exit()
    else
    end
end
```

Listing 7.2: Matlab code for automating 12 runs of 12 sets of validation images

7.1.3 Analysis and Visualisation of Feature Ratings Correlation

Concept correlation between C2 (concept bottleneck model) and hybrid network () This code averages each validation set e.g all 12 instances of rock validation numbers 1,2,3

In [ ]:
```
import os
import pandas as pd
import fnmatch
import pandoc

root = "C:/Users/c21012241/Dropbox"

### 13 Features
path =  root + "/13 Features - Binary Crystals/\
C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 13U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path =  root + "13 Features - Continuous Crystals/\
#C2 LR 10^-3 Epochs 200 MiniBatch 1024 - Hybrid LR 10^-3 Epochs 50 MiniBatch 1024 - 12 of 12"

#path =  root + "/13 Features - Binary Crystals/\
#C2 LR10^-3 E200 MB1025 - H LR10^-3 E15 MB1024 - 13U LR10^-3 E200 MB1024 - 12 of 12"

### 12 Features
#path =  root +"/12 Features - Binary Crystals + No Brightness/\
#C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 12U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path =  root +"/12 Features - Continuous Crystals + No Bright/\
#C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024 - 12U LR10^-3 E200 MB1024 - 12of12"

### Trained with re-rated features dataset
#path =  root +"/Re-rated expertFeatures - 13 - Binary/\
#C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 13U LR 10^-3 E 200 MB 1024 - 12 of 12"
```

In [ ]:
```
C2_Predicted_Features = []
hybrid_13_Nodes = []
Val_C2_1_2_3 = []
Val_C2_4_5_6 = []
Val_C2_7_8_9 = []
Val_C2_10_11_12 = []
Val_C2_1_4_7 = []
Val_C2_5_8_10 = []
Val_C2_2_9_11 = []
Val_C2_3_6_12 = []
Val_C2_1_6_9 = []
Val_C2_2_7_10 = []
Val_C2_3_8_11 = []
Val_C2_4_5_12 = []
Val_Hybrid_1_2_3 = []
Val_Hybrid_4_5_6 = []
Val_Hybrid_7_8_9 = []
Val_Hybrid_10_11_12 = []
Val_Hybrid_1_4_7 = []
Val_Hybrid_5_8_10 = []
Val_Hybrid_2_9_11 = []
Val_Hybrid_3_6_12 = []
Val_Hybrid_1_6_9 = []
Val_Hybrid_2_7_10 = []
Val_Hybrid_3_8_11 = []
Val_Hybrid_4_5_12 = []
```

In [ ]:
```
keyword_C2 = "*C2 Network - predicted_features*"
keyword_Hybrid = "*netHybrid - Average 13 Node Activations of 13x256 matrix*"

#keyword_Hybrid = "*netHybrid-Av13NodeActs*"
#keyword_C2 = "*C2-PredFeatures*"
```

In [ ]:
```
keyword_01_02_03 = "*Val_1 2 3*"
keyword_04_05_06 = "*Val_4 5 6*"
keyword_07_08_09 = "*Val_7 8 9*"
keyword_10_11_12 = "*Val_10 11 12*"
keyword_01_04_07 = "*Val_1 4 7*"
keyword_05_08_10 = "*Val_5 8 10*"
keyword_02_09_11 = "*Val_2 9 11*"
keyword_03_06_12 = "*Val_3 6 12*"
keyword_01_06_09 = "*Val_1 6 9*"
keyword_02_07_10 = "*Val_2 7 10*"
keyword_03_08_11 = "*Val_3 8 11*"
keyword_04_05_12 = "*Val_4 5 12*"
```

In [ ]:
```
# Walk through the root folder into sub folders
for root, dirs, files in os.walk(path):
# If a file name matches the C2 keyword, add it to the list
    for filename in fnmatch.filter(files, keyword_C2):
        file_path = os.path.join(root, filename)
        C2_Predicted_Features.append(file_path)

# Walk through the root folder into sub folders
for root, dirs, files in os.walk(path):
# If a file name matches hybrid network keyword, add it to the list
    for filename in fnmatch.filter(files, keyword_Hybrid):
        file_path = os.path.join(root, filename)
        hybrid_13_Nodes.append(file_path)

# Sort the list based on the time stamp
C2_Predicted_Features.sort(key=os.path.getmtime)
hybrid_13_Nodes.sort(key=os.path.getmtime)
```

In [ ]:

```
# Walk through the sorted list and if a keyword matches then add it to the relevant list
for file in C2_Predicted_Features:
    if fnmatch.fnmatch(file, keyword_01_02_03):
        df = pd.read_csv(file, header=None)
        Val_C2_1_2_3.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_06):
        df = pd.read_csv(file, header=None)
        Val_C2_4_5_6.append(df)
    elif fnmatch.fnmatch(file, keyword_07_08_09):
        df = pd.read_csv(file, header=None)
        Val_C2_7_8_9.append(df)
    elif fnmatch.fnmatch(file, keyword_10_11_12):
        df = pd.read_csv(file, header=None)
        Val_C2_10_11_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_04_07):
        df = pd.read_csv(file, header=None)
        Val_C2_1_4_7.append(df)
    elif fnmatch.fnmatch(file, keyword_05_08_10):
        df = pd.read_csv(file, header=None)
        Val_C2_5_8_10.append(df)
    elif fnmatch.fnmatch(file, keyword_02_09_11):
        df = pd.read_csv(file, header=None)
        Val_C2_2_9_11.append(df)
    elif fnmatch.fnmatch(file, keyword_03_06_12):
        df = pd.read_csv(file, header=None)
        Val_C2_3_6_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_06_09):
        df = pd.read_csv(file, header=None)
        Val_C2_1_6_9.append(df)
    elif fnmatch.fnmatch(file, keyword_02_07_10):
        df = pd.read_csv(file, header=None)
        Val_C2_2_7_10.append(df)
    elif fnmatch.fnmatch(file, keyword_03_08_11):
        df = pd.read_csv(file, header=None)
        Val_C2_3_8_11.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_12):
        df = pd.read_csv(file, header=None)
        Val_C2_4_5_12.append(df)
```

In [ ]:

```
for file in hybrid_13_Nodes:
    if fnmatch.fnmatch(file, keyword_01_02_03):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_1_2_3.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_06):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_4_5_6.append(df)
    elif fnmatch.fnmatch(file, keyword_07_08_09):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_7_8_9.append(df)
    elif fnmatch.fnmatch(file, keyword_10_11_12):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_10_11_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_04_07):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_1_4_7.append(df)
    elif fnmatch.fnmatch(file, keyword_05_08_10):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_5_8_10.append(df)
    elif fnmatch.fnmatch(file, keyword_02_09_11):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_2_9_11.append(df)
    elif fnmatch.fnmatch(file, keyword_03_06_12):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_3_6_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_06_09):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_1_6_9.append(df)
    elif fnmatch.fnmatch(file, keyword_02_07_10):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_2_7_10.append(df)
    elif fnmatch.fnmatch(file, keyword_03_08_11):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_3_8_11.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_12):
        df = pd.read_csv(file, header=None)
        Val_Hybrid_4_5_12.append(df)
```

In [ ]:

```
# C2
# Sum and average each validation set, then save to a csv file
Av_C2_Val_1_2_3 = sum(Val_C2_1_2_3)/12
Av_C2_Val_1_2_3.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_01_02_03.replace("*","") +'.csv',
                       header=None, index = False, encoding='utf-8')
Av_C2_Val_4_5_6 = sum(Val_C2_4_5_6)/12
Av_C2_Val_4_5_6.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_04_05_06.replace("*","") +'.csv',
                       header=None, index = False, encoding='utf-8')
Av_C2_Val_7_8_9 = sum(Val_C2_7_8_9)/12
Av_C2_Val_7_8_9.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_07_08_09.replace("*","") +'.csv',
                       header=None, index = False, encoding='utf-8')
Av_C2_Val_10_11_12 = sum(Val_C2_10_11_12)/12
Av_C2_Val_10_11_12.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_10_11_12.replace("*","") +'.csv',
                          header=None, index = False, encoding='utf-8')
Av_C2_Val_1_4_7 = sum(Val_C2_1_4_7)/12
Av_C2_Val_1_4_7.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_01_04_07.replace("*","") +'.csv',
                       header=None, index = False, encoding='utf-8')
Av_C2_Val_5_8_10 = sum(Val_C2_5_8_10)/12
Av_C2_Val_5_8_10.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_05_08_10.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
Av_C2_Val_2_9_11 = sum(Val_C2_2_9_11)/12
Av_C2_Val_2_9_11.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_02_09_11.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
Av_C2_Val_3_6_12 = sum(Val_C2_3_6_12)/12
Av_C2_Val_3_6_12.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_03_06_12.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
Av_C2_Val_1_6_9 = sum(Val_C2_1_6_9)/12
Av_C2_Val_1_6_9.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_01_06_09.replace("*","") +'.csv',
                       header=None, index = False, encoding='utf-8')
Av_C2_Val_2_7_10 = sum(Val_C2_2_7_10)/12
Av_C2_Val_2_7_10.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_02_07_10.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
Av_C2_Val_3_8_11 = sum(Val_C2_3_8_11)/12
Av_C2_Val_3_8_11.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_03_08_11.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
Av_C2_Val_4_5_12 = sum(Val_C2_4_5_12)/12
Av_C2_Val_4_5_12.to_csv(path + " " + keyword_C2.replace("*","") + " " + keyword_04_05_12.replace("*","") +'.csv',
                        header=None, index = False, encoding='utf-8')
```

In [ ]:

```
# Hybrid
# Sum and average each validation set, then save to a csv file
Av_Hybrid_Val_1_2_3 = sum(Val_Hybrid_1_2_3)/12
Av_Hybrid_Val_1_2_3.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_01_02_03.replace("*","") +'.csv',
                           header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_4_5_6 = sum(Val_Hybrid_4_5_6)/12
Av_Hybrid_Val_4_5_6.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_04_05_06.replace("*","") +'.csv',
                           header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_7_8_9 = sum(Val_Hybrid_7_8_9)/12
Av_Hybrid_Val_7_8_9.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_07_08_09.replace("*","") +'.csv',
                           header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_10_11_12 = sum(Val_Hybrid_10_11_12)/12
Av_Hybrid_Val_10_11_12.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_10_11_12.replace("*","") +'.csv',
                              header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_1_4_7 = sum(Val_Hybrid_1_4_7)/12
Av_Hybrid_Val_1_4_7.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_01_04_07.replace("*","") +'.csv',
                           header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_5_8_10 = sum(Val_Hybrid_5_8_10)/12
Av_Hybrid_Val_5_8_10.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_05_08_10.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_2_9_11 = sum(Val_Hybrid_2_9_11)/12
Av_Hybrid_Val_2_9_11.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_02_09_11.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_3_6_12 = sum(Val_Hybrid_3_6_12)/12
Av_Hybrid_Val_3_6_12.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_03_06_12.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_1_6_9 = sum(Val_Hybrid_1_6_9)/12
Av_Hybrid_Val_1_6_9.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_01_06_09.replace("*","") +'.csv',
                           header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_2_7_10 = sum(Val_Hybrid_2_7_10)/12
Av_Hybrid_Val_2_7_10.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_02_07_10.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_3_8_11 = sum(Val_Hybrid_3_8_11)/12
Av_Hybrid_Val_3_8_11.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_03_08_11.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
Av_Hybrid_Val_4_5_12 = sum(Val_Hybrid_4_5_12)/12
Av_Hybrid_Val_4_5_12.to_csv(path + " " + keyword_Hybrid.replace("*","") + " " + keyword_04_05_12.replace("*","") +'.csv',
                            header=None, index = False, encoding='utf-8')
```

In [ ]:

```
# A function to arrange rocks in order - order set by the original Nosofsky image folders rock order

def arrangeValidationInRockOrder(all_dfs):
    Granite = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[0:3]
        Granite = pd.concat([Granite, a], axis=0,ignore_index=True)

    Obsidian = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[3:6]
        Obsidian = pd.concat([Obsidian, a], axis=0,ignore_index=True)

    Pegmatite = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[6:9]
        Pegmatite = pd.concat([Pegmatite, a], axis=0,ignore_index=True)

    Pumice = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[9:12]
        Pumice = pd.concat([Pumice, a], axis=0,ignore_index=True)

    Gneiss = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[12:15]
        Gneiss = pd.concat([Gneiss, a], axis=0,ignore_index=True)

    Marble = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[15:18]
        Marble = pd.concat([Marble, a], axis=0,ignore_index=True)

    Slate = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[18:21]
        Slate = pd.concat([Slate, a], axis=0,ignore_index=True)

    Breccia = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[21:24]
        Breccia = pd.concat([Breccia, a], axis=0,ignore_index=True)

    Conglomerate = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[24:27]
        Conglomerate = pd.concat([Conglomerate, a], axis=0,ignore_index=True)

    Sandstone = pd.DataFrame()
    for df in all_dfs:
        a = df.iloc[27:]
        Sandstone = pd.concat([Sandstone, a], axis=0,ignore_index=True)

    return Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone
```

In [ ]:

```
# A function to append four data sets in user defined order

def appendAverageDfs(valSet1, valSet2, valSet3, valSet4):
    all_Arrange = []
    all_Arrange.append(valSet1)
    all_Arrange.append(valSet2)
    all_Arrange.append(valSet3)
    all_Arrange.append(valSet4)
    return all_Arrange
```

In [ ]:

```
# Arrangement 1
# C2
all_Ar1_C2_Dfs = appendAverageDfs(Av_C2_Val_1_2_3, Av_C2_Val_4_5_6, Av_C2_Val_7_8_9, Av_C2_Val_10_11_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar1_C2_Dfs)

all_Rocks_Ar1_C2 = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar1_C2.to_csv(path + '_all_Rocks_Ar1_C2.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:

```
# Arrangement 1
# Hybrid
all_Ar1_Hybrid_Dfs = appendAverageDfs(Av_Hybrid_Val_1_2_3, Av_Hybrid_Val_4_5_6, Av_Hybrid_Val_7_8_9, Av_Hybrid_Val_10_11_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar1_Hybrid_Dfs)

all_Rocks_Ar1_Hybrid = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar1_Hybrid.to_csv(path + '_all_Rocks_Ar1_Hybrid.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:

```
# Function to rearrange data based on new index variables for arrangement 2 & 3
def setSortIndex(rockName, index):
    rockName = rockName.set_index([index])
    rockName = rockName.sort_index()
    return rockName
```

In [ ]:
```
# Arrangement 2

# Set index as a list variable
Ar2_index = [1, 4, 7, 5, 8, 10, 2, 9, 11, 3, 6, 12 ]

# C2
all_Ar2_C2_Dfs = appendAverageDfs(Av_C2_Val_1_4_7, Av_C2_Val_5_8_10, Av_C2_Val_2_9_11, Av_C2_Val_3_6_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar2_C2_Dfs)

Granite = setSortIndex(Granite, Ar2_index)
Obsidian = setSortIndex(Obsidian, Ar2_index)
Pegmatite = setSortIndex(Pegmatite, Ar2_index)
Pumice = setSortIndex(Pumice, Ar2_index)
Gneiss = setSortIndex(Gneiss, Ar2_index)
Marble = setSortIndex(Marble, Ar2_index)
Slate = setSortIndex(Slate, Ar2_index)
Breccia = setSortIndex(Breccia, Ar2_index)
Conglomerate = setSortIndex(Conglomerate, Ar2_index)
Sandstone = setSortIndex(Sandstone, Ar2_index)

all_Rocks_Ar2_C2 = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar2_C2.to_csv(path + '_all_Rocks_Ar2_C2.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:
```
# Hybrid
all_Ar2_Hybrid_Dfs = appendAverageDfs(Av_Hybrid_Val_1_4_7, Av_Hybrid_Val_5_8_10, Av_Hybrid_Val_2_9_11, Av_Hybrid_Val_3_6_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar2_Hybrid_Dfs)

Granite = setSortIndex(Granite, Ar2_index)
Obsidian = setSortIndex(Obsidian, Ar2_index)
Pegmatite = setSortIndex(Pegmatite, Ar2_index)
Pumice = setSortIndex(Pumice, Ar2_index)
Gneiss = setSortIndex(Gneiss, Ar2_index)
Marble = setSortIndex(Marble, Ar2_index)
Slate = setSortIndex(Slate, Ar2_index)
Breccia = setSortIndex(Breccia, Ar2_index)
Conglomerate = setSortIndex(Conglomerate, Ar2_index)
Sandstone = setSortIndex(Sandstone, Ar2_index)

all_Rocks_Ar2_Hybrid = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar2_Hybrid.to_csv(path + '_all_Rocks_Ar2_Hybrid.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:
```
# Arrangement 3

# C2
all_Ar3_C2_Dfs = appendAverageDfs(Av_C2_Val_1_6_9, Av_C2_Val_2_7_10, Av_C2_Val_3_8_11, Av_C2_Val_4_5_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar3_C2_Dfs)

# Set index as a list variable
Ar3_index = [1, 6, 9, 2, 7, 10, 3, 8, 11, 4, 5, 12]

Granite = setSortIndex(Granite, Ar3_index)
Obsidian = setSortIndex(Obsidian, Ar3_index)
Pegmatite = setSortIndex(Pegmatite, Ar3_index)
Pumice = setSortIndex(Pumice, Ar3_index)
Gneiss = setSortIndex(Gneiss, Ar3_index)
Marble = setSortIndex(Marble, Ar3_index)
Slate = setSortIndex(Slate, Ar3_index)
Breccia = setSortIndex(Breccia, Ar3_index)
Conglomerate = setSortIndex(Conglomerate, Ar3_index)
Sandstone = setSortIndex(Sandstone, Ar3_index)

all_Rocks_Ar3_C2 = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar3_C2.to_csv(path + '_all_Rocks_Ar3_C2.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:
```
# Arrangement 3

# Hybrid
all_Ar3_Hybrid_Dfs = appendAverageDfs(Av_Hybrid_Val_1_6_9, Av_Hybrid_Val_2_7_10, Av_Hybrid_Val_3_8_11, Av_Hybrid_Val_4_5_12)

[Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone] = arrangeValidationInRockOrder(all_Ar3_Hybrid_Dfs)

Granite = setSortIndex(Granite, Ar3_index)
Obsidian = setSortIndex(Obsidian, Ar3_index)
Pegmatite = setSortIndex(Pegmatite, Ar3_index)
Pumice = setSortIndex(Pumice, Ar3_index)
Gneiss = setSortIndex(Gneiss, Ar3_index)
Marble = setSortIndex(Marble, Ar3_index)
Slate = setSortIndex(Slate, Ar3_index)
Breccia = setSortIndex(Breccia, Ar3_index)
Conglomerate = setSortIndex(Conglomerate, Ar3_index)
Sandstone = setSortIndex(Sandstone, Ar3_index)

all_Rocks_Ar3_Hybrid = pd.concat([Granite,Obsidian,Pegmatite,Pumice,Gneiss,Marble,Slate,Breccia,Conglomerate,Sandstone], axis=0, ignore_index=True)
all_Rocks_Ar3_Hybrid.to_csv(path + '_all_Rocks_Ar3_Hybrid.csv', header=None, index = False, encoding='utf-8')
```

In [ ]:
```
# Add the three arrangements of data together and divide by 3

# C2
All_C2_Arrangements = pd.DataFrame()
All_C2_Arrangements = all_Rocks_Ar1_C2.add(All_C2_Arrangements, fill_value=0)
All_C2_Arrangements = all_Rocks_Ar2_C2.add(All_C2_Arrangements, fill_value=0)
All_C2_Arrangements = all_Rocks_Ar3_C2.add(All_C2_Arrangements, fill_value=0)

Av_C2_Arrangements = All_C2_Arrangements/3
Av_C2_Arrangements.to_csv(path+"_Av_C2_Arrangements.csv", header=None, index = False, encoding='utf-8')
```

In [ ]:

```
# Add the three arrangements of data together and divide by 3

# Hybrid
All_Hybrid_Arrangements = pd.DataFrame()
All_Hybrid_Arrangements = all_Rocks_Ar1_Hybrid.add(All_Hybrid_Arrangements, fill_value=0)
All_Hybrid_Arrangements = all_Rocks_Ar2_Hybrid.add(All_Hybrid_Arrangements, fill_value=0)
All_Hybrid_Arrangements = all_Rocks_Ar3_Hybrid.add(All_Hybrid_Arrangements, fill_value=0)

Av_Hybrid_Arrangements = All_Hybrid_Arrangements/3
Av_Hybrid_Arrangements.to_csv(path+"_Av_Hybrid_Arrangements.csv", header=None, index = False, encoding='utf-8')
```

In [ ]:

```
### Don't forget to change! ###

root = "C:/Users/c21012241/Dropbox"

### 13 Features
pathExpert = root +"/expertRatings/expertRatings - Binary Crystals/Binary Crystals - 13 Features/\
Ratings transformed - for use in matlab visual/expertRatings.csv"
#pathExpert = root +"/expertRatings/expertRatings - Continuous Crystals/Continuous Crystals - 13 Features/expertRatings.csv"

### 12 Features

#pathExpert = root +"/expertRatings/expertRatings - Binary Crystals/Binary Crystals - 12 Features/expertRatings.csv"
#pathExpert = root +"/expertRatings/expertRatings - Continuous Crystals/Continuous Crystals - 12 Features/expertRatings.csv"


### Re-rated 13 expert features
#pathExpert = root +"/XAI_Feature_Anomyly/XAI_Feature_Anomyly/expertFeatures13Binary - Correlation Plot Set - Transformed.csv"

expertRatings = pd.read_csv(pathExpert, header = None)
```

In [ ]:

```
expertHybridCorrelation = expertRatings.corrwith(Av_Hybrid_Arrangements)
print(expertHybridCorrelation)
expertHybridCorrelation.to_csv(path+"_expertHybridCorrelation.csv", header=None, index = False, encoding='utf-8')
```

In [ ]:

```
expertC2Correlation = expertRatings.corrwith(Av_C2_Arrangements)
print(expertC2Correlation)
expertC2Correlation.to_csv(path+"_expertC2Correlation.csv", header=None, index = False, encoding='utf-8')
```

In [ ]:

```
hybridC2Correlation = Av_Hybrid_Arrangements.corrwith(Av_C2_Arrangements)
print(hybridC2Correlation)
hybridC2Correlation.to_csv(path+"_hybridC2Correlation.csv", header=None, index = False, encoding='utf-8')
```

In [ ]:

```
import plotly.graph_objects as go
import kaleido

# 13 Features

features = ['Average Grainsize','Roughness', 'Presence of Foliation', 'Presence of Banding', 'Heterogeneity of Grainsize',
'Lightness of Colour', 'Heterogeneity of Hue', 'Heterogeneity of Brightness', 'Volume of Vesicles', 'Glasslike Texture',
'Angular Clasts', 'Rounded Clasts', 'Presence of Crystals']

# 12 Features

#features = ['Average Grainsize','Roughness', 'Presence of Foliation', 'Presence of Banding', 'Heterogeneity of Grainsize',
#'Lightness of Colour', 'Heterogeneity of Hue', 'Volume of Vesicles', 'Glasslike Texture',
#'Angular Clasts', 'Rounded Clasts', 'Presence of Crystals']

fig = go.Figure()
fig.add_trace(go.Bar(
    x=features,
    y=expertHybridCorrelation,
    name='Expert vs Hybrid',
    marker_color="rgb(253,174,97)"
))
fig.add_trace(go.Bar(
    x=features,
    y=expertC2Correlation,
    name='Expert vs C2',
    marker_color="rgb(178,223,138)"
))
fig.add_trace(go.Bar(
    x=features,
    y=hybridC2Correlation,
    name='Hybrid vs C2',
    marker_color="rgb(116,173,209)"
))

title="13 Feature Concepts - Binary Crystal Rating - Correlation: C2 LR10^-3 E200 MB1025 - H LR10^-3 E50 MB1024 - 12 of 12"

fig.update_layout(barmode='group',
                    xaxis_tickangle=-45,
                    plot_bgcolor="#fff",
                    title=title,
                    xaxis_title="Rock Feature",
                    yaxis_title="Pearson Correlation Coefficient",
                font=dict(
                    family="Helvetica",
                    size=9,
                    color="Black"))
fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='LightPink')
fig.update_layout(title_pad_l=400,
                    title_pad_r=400,
                    title={
                    'text': title,
                    'y':0.9,
                    'x':0.5,
                    'xanchor': 'center',
                    'yanchor': 'top'})

fig.show()

fig.write_image(path + "/" + "Correlation - C2 vs Hybrid Networks.png")
```

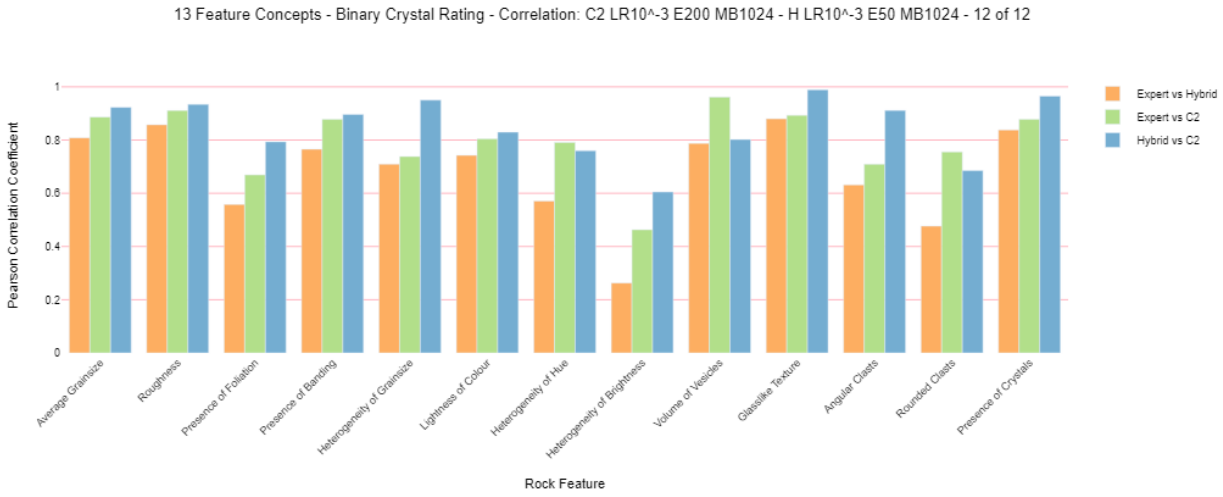


Figure 7.1: Code for the Analysis and Visualisation of Pearson's Correlation Co-Efficient of Feature Ratings Between Expert Feature Ratings, Sequential CBM and Hybrid Classifier CBM. This was coded using Python in a Jupyter Notebook, and Plotly for visualisation

7.1.4 Means and Standard Error of the Mean (SEM) - Analysis and Visualisation using Plotly

In [ ]:

```
import os
import pandas as pd
import fnmatch
import numpy as np

root = "C:/Users/c21012241/Dropbox"

### 13 Features ###
path = root + "/13 Features - Binary Crystals/\
C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 13U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path = root +"/13 Features - Continuous Crystals/\
#C2 LR 10^-3 Epochs 200 MiniBatch 1024 - Hybrid LR 10^-3 Epochs 50 MiniBatch 1024 - 12 of 12"

#path = root + "/13 Features - Continuous Crystals/\
#C2 LR 10^-3 E 150 MB 1024 - H LR 10^-3 E 15 MB 1024 - 13Un LR 10^-3 E 150 MB 1024 - 12of12"

#path = root +"/13 Features - Binary Crystals/\
#C2 LR10^-3 E200 MB1024 - H LR10^-3 E15 MB1024 - 13U LR10^-3 E200 MB1024 - 12 of 12"

### 12 Features ###
#path = root + "/12 Features - Binary Crystals + No Brightness/\
#C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 12U LR 10^-3 E 200 MB 1024 - 12 of 12"

#path = root + "/12 Features - Continuous Crystals + No Bright/\
#C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024 - 12U LR10^-3 E200 MB1024 - 12of12"
```

In [ ]:

```
rockNamesTen = ["Granite", "Obsidian", "Pegmatite", "Pumice", "Gneiss", "Marble", "Slate", "Breccia", "Conglomerate", "Sandstone"]
Val_Hybrid_1_2_3 = []
Val_Hybrid_4_5_6 = []
Val_Hybrid_7_8_9 = []
Val_Hybrid_10_11_12 = []
Val_Hybrid_1_4_7 = []
Val_Hybrid_5_8_10 = []
Val_Hybrid_2_9_11 = []
Val_Hybrid_3_6_12 = []
Val_Hybrid_1_6_9 = []
Val_Hybrid_2_7_10 = []
Val_Hybrid_3_8_11 = []
Val_Hybrid_4_5_12 = []
Val_1_2_3 = []
Val_4_5_6 = []
Val_7_8_9 = []
Val_10_11_12 = []
Val_1_4_7 = []
Val_5_8_10 = []
Val_2_9_11 = []
Val_3_6_12 = []
Val_1_6_9 = []
Val_2_7_10 = []
Val_3_8_11 = []
Val_4_5_12 = []
keyword_01_02_03 = "*Val_1 2 3*"
keyword_04_05_06 = "*Val_4 5 6*"
keyword_07_08_09 = "*Val_7 8 9*"
keyword_10_11_12 = "*Val_10 11 12*"
keyword_01_04_07 = "*Val_1 4 7*"
keyword_05_08_10 = "*Val_5 8 10*"
keyword_02_09_11 = "*Val_2 9 11*"
keyword_03_06_12 = "*Val_3 6 12*"
keyword_01_06_09 = "*Val_1 6 9*"
keyword_02_07_10 = "*Val_2 7 10*"
keyword_03_08_11 = "*Val_3 8 11*"
keyword_04_05_12 = "*Val_4 5 12*"
keywordConfusion = '*Confusion*_Matrix*'
all_Confusion = []
C2_Accuracy = []
hybrid_Accuracy = []
all_Confusion_DF = []
```

In [ ]:

```
#Get all confusion matrix adn append to all_Confusion
for root, dirs, files in os.walk(path):
    for filename in fnmatch.filter(files, keywordConfusion):
        file_path = os.path.join(root, filename)
        all_Confusion.append(file_path)

# Sort all by date
all_Confusion.sort(key=os.path.getmtime)

# Walk through the sorted list and if a validation set keyword matches then add it to the relevant list
for file in all_Confusion:
    if fnmatch.fnmatch(file, keyword_01_02_03):
        df = pd.read_csv(file, header=None)
        Val_1_2_3.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_06):
        df = pd.read_csv(file, header=None)
        Val_4_5_6.append(df)
    elif fnmatch.fnmatch(file, keyword_07_08_09):
        df = pd.read_csv(file, header=None)
        Val_7_8_9.append(df)
    elif fnmatch.fnmatch(file, keyword_10_11_12):
        df = pd.read_csv(file, header=None)
        Val_10_11_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_04_07):
        df = pd.read_csv(file, header=None)
        Val_1_4_7.append(df)
    elif fnmatch.fnmatch(file, keyword_05_08_10):
        df = pd.read_csv(file, header=None)
        Val_5_8_10.append(df)
    elif fnmatch.fnmatch(file, keyword_02_09_11):
        df = pd.read_csv(file, header=None)
        Val_2_9_11.append(df)
    elif fnmatch.fnmatch(file, keyword_03_06_12):
        df = pd.read_csv(file, header=None)
        Val_3_6_12.append(df)
    elif fnmatch.fnmatch(file, keyword_01_06_09):
        df = pd.read_csv(file, header=None)
        Val_1_6_9.append(df)
    elif fnmatch.fnmatch(file, keyword_02_07_10):
        df = pd.read_csv(file, header=None)
        Val_2_7_10.append(df)
    elif fnmatch.fnmatch(file, keyword_03_08_11):
        df = pd.read_csv(file, header=None)
        Val_3_8_11.append(df)
    elif fnmatch.fnmatch(file, keyword_04_05_12):
        df = pd.read_csv(file, header=None)
        Val_4_5_12.append(df)

for file in all_Confusion:
    if fnmatch.fnmatch(file, keywordConfusion):
        df = pd.read_csv(file, header=None)
        all_Confusion_DF.append(df)
```

In [ ]:

```
def sumConfusionDfC2(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 0:10]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion

def sumConfusionDfHybrid(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 10:20]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion

def sumConfusionDfUnconstrained(Confusion, all_Confusion_DF):
    for df in all_Confusion_DF:
        df = df.apply(pd.to_numeric, errors='coerce')
        a = df.iloc[2:12, 20:30]
        Confusion = Confusion.add(a, fill_value=0)
    return Confusion
```

In [ ]:

```
C2Confusion = pd.DataFrame()
C2Confusion = sumConfusionDfC2(C2Confusion, all_Confusion_DF)
C2Confusion = C2Confusion.reset_index(drop=True)
C2Confusion.index = rockNamesTen
C2Confusion.columns = rockNamesTen
file_path = path + "/C2Confusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

C2Confusion.to_csv(new_file_path)
```

In [ ]:

```
import plotly.graph_objects as go

total_sum_C2_Confusion = np.sum(C2Confusion.values)
C2ConfusionPercentages = np.round(((C2Confusion.values/total_sum_C2_Confusion) * 1000),2)

z = C2ConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "C2 Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB1024 - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}

}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "C2 Confusion Matrix.png")
```

In [ ]:

```
HybridConfusion = pd.DataFrame()
HybridConfusion = sumConfusionDfHybrid(HybridConfusion, all_Confusion_DF)
HybridConfusion = HybridConfusion.reset_index(drop=True)
HybridConfusion.index = rockNamesTen
HybridConfusion.columns = rockNamesTen
file_path = path + "/HybridConfusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

HybridConfusion.to_csv(new_file_path)
```

In [ ]:

```
import plotly.graph_objects as go

total_sum_Hybrid_Confusion = np.sum(HybridConfusion.values)
HybridConfusionPercentages = np.round(((HybridConfusion.values/total_sum_Hybrid_Confusion) * 1000),2)

z = HybridConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "Hybrid Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB1025 - H LR10^-3 E15 MB1024 - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}
}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "Hybrid Confusion Matrix.png")
```

In [ ]:

```
UnconstrainedConfusion = pd.DataFrame()
UnconstrainedConfusion = sumConfusionDfUnconstrained(UnconstrainedConfusion, all_Confusion_DF)
UnconstrainedConfusion = UnconstrainedConfusion.reset_index(drop=True)
UnconstrainedConfusion.index = rockNamesTen
UnconstrainedConfusion.columns = rockNamesTen
file_path = path + "/UnconstrainedConfusion" + ".csv"

if os.path.isfile(file_path):
    # If the file already exists, create a new one with "_1" appended
    root, ext = os.path.splitext(file_path)
    new_file_path = root + "_1" + ext
else:
    # If not, use the original file path
    new_file_path = file_path

UnconstrainedConfusion.to_csv(new_file_path)
```

In [ ]:

```
import plotly.graph_objects as go

total_sum_Hybrid_Confusion = np.sum(UnconstrainedConfusion.values)
HybridConfusionPercentages = np.round(((HybridConfusion.values/total_sum_Hybrid_Confusion) * 1000),2)

z = HybridConfusionPercentages
x = rockNamesTen
y = rockNamesTen
z_text = [[str(y) for y in x] for x in z]

layout = {
        "title": "Hybrid Confusion Matrix - 13 Features - Binary Crystals - C2 LR10^-3 E200 MB102512 Runs of 12 Alternating Validation Images - 12 of 12",
        "xaxis": {"title": "Predicted value"},
        "yaxis": {"title": "Real value"}

}

fig = go.Figure(data=go.Heatmap(z=z, x=x, y=y, autocolorscale = False,
                                colorscale = [[0, 'rgb(255,255,255)'], [1, 'rgb(100,149,237)']],
                                hoverongaps = False), layout=layout)

# Add annotations
for i in range(len(y)):
    for j in range(len(x)):
        fig.add_annotation(
            text=str(z_text[i][j] + "%"),
            x=x[j],
            y=y[i],
            showarrow=False,
            font=dict(size=12),
            visible=True,
            xanchor='center',
            yanchor='middle'
        )

fig.show()
fig.write_image(path + "/" + "Hybrid Confusion Matrix.png")
```

In [ ]:

```
# Function to split hybrid and C2 networks accuracies in
def splitAccuraciesToDfC2(constrainedC2ValAcc, validationSet):
    for df in validationSet:
        a = df.iloc[12,1:2]
        constrainedC2ValAcc = pd.concat([constrainedC2ValAcc, a], axis=0,ignore_index=True)
    return constrainedC2ValAcc

def splitAccuraciesToDfHybrid(hybridNetworkValAcc, validationSet):
    for df in validationSet:
        a = df.iloc[12,11:12]
        hybridNetworkValAcc = pd.concat([hybridNetworkValAcc, a], axis=0,ignore_index=True)
    return hybridNetworkValAcc
```

In [ ]:

```
Val_Acc_C2_1_2_3 = pd.DataFrame()
Val_Acc_C2_1_2_3 = splitAccuraciesToDfC2(Val_Acc_C2_1_2_3, Val_1_2_3)
Val_Acc_C2_4_5_6 = pd.DataFrame()
Val_Acc_C2_4_5_6 = splitAccuraciesToDfC2(Val_Acc_C2_4_5_6, Val_4_5_6)
Val_Acc_C2_7_8_9 = pd.DataFrame()
Val_Acc_C2_7_8_9 = splitAccuraciesToDfC2(Val_Acc_C2_7_8_9, Val_7_8_9)
Val_Acc_C2_10_11_12 = pd.DataFrame()
Val_Acc_C2_10_11_12 = splitAccuraciesToDfC2(Val_Acc_C2_10_11_12, Val_10_11_12)
Val_Acc_C2_1_4_7 = pd.DataFrame()
Val_Acc_C2_1_4_7 = splitAccuraciesToDfC2(Val_Acc_C2_1_4_7, Val_1_4_7)
Val_Acc_C2_5_8_10 = pd.DataFrame()
Val_Acc_C2_5_8_10 = splitAccuraciesToDfC2(Val_Acc_C2_5_8_10, Val_5_8_10)
Val_Acc_C2_2_9_11 = pd.DataFrame()
Val_Acc_C2_2_9_11 = splitAccuraciesToDfC2(Val_Acc_C2_2_9_11, Val_2_9_11)
Val_Acc_C2_3_6_12 = pd.DataFrame()
Val_Acc_C2_3_6_12 = splitAccuraciesToDfC2(Val_Acc_C2_3_6_12, Val_3_6_12)
Val_Acc_C2_1_6_9 = pd.DataFrame()
Val_Acc_C2_1_6_9 = splitAccuraciesToDfC2(Val_Acc_C2_1_6_9, Val_1_6_9)
Val_Acc_C2_2_7_10 = pd.DataFrame()
Val_Acc_C2_2_7_10 = splitAccuraciesToDfC2(Val_Acc_C2_2_7_10, Val_2_7_10)
Val_Acc_C2_3_8_11 = pd.DataFrame()
Val_Acc_C2_3_8_11 = splitAccuraciesToDfC2(Val_Acc_C2_3_8_11, Val_3_8_11)
Val_Acc_C2_4_5_12 = pd.DataFrame()
Val_Acc_C2_4_5_12 = splitAccuraciesToDfC2(Val_Acc_C2_4_5_12, Val_4_5_12)

Val_Acc_Hybrid_1_2_3 = pd.DataFrame()
Val_Acc_Hybrid_1_2_3 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_2_3, Val_1_2_3)
Val_Acc_Hybrid_4_5_6 = pd.DataFrame()
Val_Acc_Hybrid_4_5_6 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_4_5_6, Val_4_5_6)
Val_Acc_Hybrid_7_8_9 = pd.DataFrame()
Val_Acc_Hybrid_7_8_9 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_7_8_9, Val_7_8_9)
Val_Acc_Hybrid_10_11_12 = pd.DataFrame()
Val_Acc_Hybrid_10_11_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_10_11_12, Val_10_11_12)
Val_Acc_Hybrid_1_4_7 = pd.DataFrame()
Val_Acc_Hybrid_1_4_7 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_4_7, Val_1_4_7)
Val_Acc_Hybrid_5_8_10 = pd.DataFrame()
Val_Acc_Hybrid_5_8_10 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_5_8_10, Val_5_8_10)
Val_Acc_Hybrid_2_9_11 = pd.DataFrame()
Val_Acc_Hybrid_2_9_11 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_2_9_11, Val_2_9_11)
Val_Acc_Hybrid_3_6_12 = pd.DataFrame()
Val_Acc_Hybrid_3_6_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_3_6_12, Val_3_6_12)
Val_Acc_Hybrid_1_6_9 = pd.DataFrame()
Val_Acc_Hybrid_1_6_9 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_1_6_9, Val_1_6_9)
Val_Acc_Hybrid_2_7_10 = pd.DataFrame()
Val_Acc_Hybrid_2_7_10 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_2_7_10, Val_2_7_10)
Val_Acc_Hybrid_3_8_11 = pd.DataFrame()
Val_Acc_Hybrid_3_8_11 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_3_8_11, Val_3_8_11)
Val_Acc_Hybrid_4_5_12 = pd.DataFrame()
Val_Acc_Hybrid_4_5_12 = splitAccuraciesToDfHybrid(Val_Acc_Hybrid_4_5_12, Val_4_5_12)
```

In [ ]:

```
def meanValStdSem(valSet):
    valSet_means = np.mean((valSet.sum(axis=1)).to_numpy())
    valSet_std = (valSet.sum(axis=1)).to_numpy().std()
    valSet_sem = valSet_std / np.sqrt(np.size(valSet))

    return valSet_means, valSet_std, valSet_sem
```

In [ ]:

```
# Validation sets mean, standard deviation and standard error of the mean

totalVal_Acc_C2_1_2_3_mean_std_sem = meanValStdSem(Val_Acc_C2_1_2_3)
totalVal_Acc_C2_4_5_6_mean_std_sem = meanValStdSem(Val_Acc_C2_4_5_6)
totalVal_Acc_C2_7_8_9_mean_std_sem = meanValStdSem(Val_Acc_C2_7_8_9)
totalVal_Acc_C2_10_11_12_mean_std_sem = meanValStdSem(Val_Acc_C2_10_11_12)
totalVal_Acc_C2_1_4_7_mean_std_sem = meanValStdSem(Val_Acc_C2_1_4_7)
totalVal_Acc_C2_5_8_10_mean_std_sem = meanValStdSem(Val_Acc_C2_5_8_10)
totalVal_Acc_C2_2_9_11_mean_std_sem = meanValStdSem(Val_Acc_C2_2_9_11)
totalVal_Acc_C2_3_6_12_mean_std_sem = meanValStdSem(Val_Acc_C2_3_6_12)
totalVal_Acc_C2_1_6_9_mean_std_sem = meanValStdSem(Val_Acc_C2_1_6_9)
totalVal_Acc_C2_2_7_10_mean_std_sem = meanValStdSem(Val_Acc_C2_2_7_10)
totalVal_Acc_C2_3_8_11_mean_std_sem = meanValStdSem(Val_Acc_C2_3_8_11)
totalVal_Acc_C2_4_5_12_mean_std_sem = meanValStdSem(Val_Acc_C2_4_5_12)
```

In [ ]:

```
totalVal_Acc_hybrid_1_2_3_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_2_3)
totalVal_Acc_hybrid_4_5_6_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_4_5_6)
totalVal_Acc_hybrid_7_8_9_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_7_8_9)
totalVal_Acc_hybrid_10_11_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_10_11_12)
totalVal_Acc_hybrid_1_4_7_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_4_7)
totalVal_Acc_hybrid_5_8_10_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_5_8_10)
totalVal_Acc_hybrid_2_9_11_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_2_9_11)
totalVal_Acc_hybrid_3_6_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_3_6_12)
totalVal_Acc_hybrid_1_6_9_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_1_6_9)
totalVal_Acc_hybrid_2_7_10_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_2_7_10)
totalVal_Acc_hybrid_3_8_11_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_3_8_11)
totalVal_Acc_hybrid_4_5_12_mean_std_sem = meanValStdSem(Val_Acc_Hybrid_4_5_12)
```

In [ ]:

```
meanOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [0] + totalVal_Acc_C2_4_5_6_mean_std_sem [0] +
                 totalVal_Acc_C2_7_8_9_mean_std_sem [0] + totalVal_Acc_C2_10_11_12_mean_std_sem [0] +
                 totalVal_Acc_C2_1_4_7_mean_std_sem [0] + totalVal_Acc_C2_5_8_10_mean_std_sem [0] +
                 totalVal_Acc_C2_2_9_11_mean_std_sem [0] + totalVal_Acc_C2_3_6_12_mean_std_sem [0] +
                 totalVal_Acc_C2_1_6_9_mean_std_sem [0] + totalVal_Acc_C2_2_7_10_mean_std_sem [0] +
                 totalVal_Acc_C2_3_8_11_mean_std_sem [0] + totalVal_Acc_C2_4_5_12_mean_std_sem [0])/12

stdOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [1] + totalVal_Acc_C2_4_5_6_mean_std_sem [1] +
                totalVal_Acc_C2_7_8_9_mean_std_sem [1] + totalVal_Acc_C2_10_11_12_mean_std_sem [1] +
                totalVal_Acc_C2_1_4_7_mean_std_sem [1] + totalVal_Acc_C2_5_8_10_mean_std_sem [1] +
                totalVal_Acc_C2_2_9_11_mean_std_sem [1] + totalVal_Acc_C2_3_6_12_mean_std_sem [1] +
                totalVal_Acc_C2_1_6_9_mean_std_sem [1] + totalVal_Acc_C2_2_7_10_mean_std_sem [1] +
                totalVal_Acc_C2_3_8_11_mean_std_sem [1] + totalVal_Acc_C2_4_5_12_mean_std_sem [1])/12

semOfMeansC2 = (totalVal_Acc_C2_1_2_3_mean_std_sem [2] + totalVal_Acc_C2_4_5_6_mean_std_sem [2] +
                totalVal_Acc_C2_7_8_9_mean_std_sem [2] + totalVal_Acc_C2_10_11_12_mean_std_sem [2] +
                totalVal_Acc_C2_1_4_7_mean_std_sem [2] + totalVal_Acc_C2_5_8_10_mean_std_sem [2] +
                totalVal_Acc_C2_2_9_11_mean_std_sem [2] + totalVal_Acc_C2_3_6_12_mean_std_sem [2] +
                totalVal_Acc_C2_1_6_9_mean_std_sem [2] + totalVal_Acc_C2_2_7_10_mean_std_sem [2] +
                totalVal_Acc_C2_3_8_11_mean_std_sem [2] + totalVal_Acc_C2_4_5_12_mean_std_sem [2])/12

meanOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [0] , totalVal_Acc_C2_4_5_6_mean_std_sem [0] ,
                    totalVal_Acc_C2_7_8_9_mean_std_sem [0] , totalVal_Acc_C2_10_11_12_mean_std_sem [0] ,
                    totalVal_Acc_C2_1_4_7_mean_std_sem [0] , totalVal_Acc_C2_5_8_10_mean_std_sem [0] ,
                    totalVal_Acc_C2_2_9_11_mean_std_sem [0] , totalVal_Acc_C2_3_6_12_mean_std_sem [0] ,
                    totalVal_Acc_C2_1_6_9_mean_std_sem [0] , totalVal_Acc_C2_2_7_10_mean_std_sem [0] ,
                    totalVal_Acc_C2_3_8_11_mean_std_sem [0] , totalVal_Acc_C2_4_5_12_mean_std_sem [0]
                   ]

stdOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [1] , totalVal_Acc_C2_4_5_6_mean_std_sem [1] ,
                   totalVal_Acc_C2_7_8_9_mean_std_sem [1] , totalVal_Acc_C2_10_11_12_mean_std_sem [1] ,
                   totalVal_Acc_C2_1_4_7_mean_std_sem [1] , totalVal_Acc_C2_5_8_10_mean_std_sem [1] ,
                   totalVal_Acc_C2_2_9_11_mean_std_sem [1] , totalVal_Acc_C2_3_6_12_mean_std_sem [1] ,
                   totalVal_Acc_C2_1_6_9_mean_std_sem [1] , totalVal_Acc_C2_2_7_10_mean_std_sem [1] ,
                   totalVal_Acc_C2_3_8_11_mean_std_sem [1] , totalVal_Acc_C2_4_5_12_mean_std_sem [1]
                  ]

semOfMeansC2_12 = [totalVal_Acc_C2_1_2_3_mean_std_sem [2] , totalVal_Acc_C2_4_5_6_mean_std_sem [2] ,
                   totalVal_Acc_C2_7_8_9_mean_std_sem [2] , totalVal_Acc_C2_10_11_12_mean_std_sem [2] ,
                   totalVal_Acc_C2_1_4_7_mean_std_sem [2] , totalVal_Acc_C2_5_8_10_mean_std_sem [2] ,
                   totalVal_Acc_C2_2_9_11_mean_std_sem [2] , totalVal_Acc_C2_3_6_12_mean_std_sem [2] ,
                   totalVal_Acc_C2_1_6_9_mean_std_sem [2] , totalVal_Acc_C2_2_7_10_mean_std_sem [2] ,
                   totalVal_Acc_C2_3_8_11_mean_std_sem [2] , totalVal_Acc_C2_4_5_12_mean_std_sem [2]
                  ]
```

In [ ]:

```
print("meanOfMeansC2 = " + str(meanOfMeansC2))
print("semOfMeansC2 = " + str(semOfMeansC2))
```

In [ ]:

```
meanOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [0] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [0] +
                     totalVal_Acc_hybrid_7_8_9_mean_std_sem [0] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [0] +
                     totalVal_Acc_hybrid_1_4_7_mean_std_sem [0] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [0] +
                     totalVal_Acc_hybrid_2_9_11_mean_std_sem [0] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [0] +
                     totalVal_Acc_hybrid_1_6_9_mean_std_sem [0] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [0] +
                     totalVal_Acc_hybrid_3_8_11_mean_std_sem [0] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [0])/12

stdOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [1] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [1] +
                    totalVal_Acc_hybrid_7_8_9_mean_std_sem [1] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [1] +
                    totalVal_Acc_hybrid_1_4_7_mean_std_sem [1] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [1] +
                    totalVal_Acc_hybrid_2_9_11_mean_std_sem [1] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [1] +
                    totalVal_Acc_hybrid_1_6_9_mean_std_sem [1] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [1] +
                    totalVal_Acc_hybrid_3_8_11_mean_std_sem [1] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [1])/12

semOfMeansHybrid = (totalVal_Acc_hybrid_1_2_3_mean_std_sem [2] + totalVal_Acc_hybrid_4_5_6_mean_std_sem [2] +
                    totalVal_Acc_hybrid_7_8_9_mean_std_sem [2] + totalVal_Acc_hybrid_10_11_12_mean_std_sem [2] +
                    totalVal_Acc_hybrid_1_4_7_mean_std_sem [2] + totalVal_Acc_hybrid_5_8_10_mean_std_sem [2] +
                    totalVal_Acc_hybrid_2_9_11_mean_std_sem [2] + totalVal_Acc_hybrid_3_6_12_mean_std_sem [2] +
                    totalVal_Acc_hybrid_1_6_9_mean_std_sem [2] + totalVal_Acc_hybrid_2_7_10_mean_std_sem [2] +
                    totalVal_Acc_hybrid_3_8_11_mean_std_sem [2] + totalVal_Acc_hybrid_4_5_12_mean_std_sem [2])/12

meanOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [0] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_7_8_9_mean_std_sem [0] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_1_4_7_mean_std_sem [0] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_2_9_11_mean_std_sem [0] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_1_6_9_mean_std_sem [0] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [0] ,
                        totalVal_Acc_hybrid_3_8_11_mean_std_sem [0] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [0]
                       ]

stdOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [1] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_7_8_9_mean_std_sem [1] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_1_4_7_mean_std_sem [1] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_2_9_11_mean_std_sem [1] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_1_6_9_mean_std_sem [1] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [1] ,
                       totalVal_Acc_hybrid_3_8_11_mean_std_sem [1] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [1]
                      ]

semOfMeansHybrid_12 = [totalVal_Acc_hybrid_1_2_3_mean_std_sem [2] , totalVal_Acc_hybrid_4_5_6_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_7_8_9_mean_std_sem [2] , totalVal_Acc_hybrid_10_11_12_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_1_4_7_mean_std_sem [2] , totalVal_Acc_hybrid_5_8_10_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_2_9_11_mean_std_sem [2] , totalVal_Acc_hybrid_3_6_12_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_1_6_9_mean_std_sem [2] , totalVal_Acc_hybrid_2_7_10_mean_std_sem [2] ,
                       totalVal_Acc_hybrid_3_8_11_mean_std_sem [2] , totalVal_Acc_hybrid_4_5_12_mean_std_sem [2]
                      ]
```

In [ ]:

```
print("meanOfMeansC2 = " + str(meanOfMeansHybrid))
print("semOfMeansC2 = " + str(semOfMeansHybrid))
```

In [ ]:

```
print("Val 3, 6, 12 " + " C2 mean = " + str(meanOfMeansC2_12[7]) + " C2 SEM = " + str(semOfMeansC2_12[7]))
print("Val 3, 6, 12 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[7]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[7]))
```

In [ ]:

```
print("Val 1, 6, 9 " + " C2 mean = " + str(meanOfMeansC2_12[8]) + " C2 SEM = " + str(semOfMeansC2_12[8]))
print("Val 1, 6, 9 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[8]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[8]))
```

In [ ]:
```
print("Val 2, 7, 10 " + " C2 mean = " + str(meanOfMeansC2_12[9]) + " C2 SEM = " + str(semOfMeansC2_12[9]))
print("Val 2, 7, 10 " + " Hybrid mean = " + str(meanOfMeansHybrid_12[9]) + " Hybrid SEM = " + str(semOfMeansHybrid_12[9]))
```

In [ ]:
```
import plotly.graph_objects as go
import kaleido

validationSets = ['Val 1, 2, 3','Val 4, 5, 6', 'Val 7, 8, 9', 'Val 10, 11, 12',
                  'Val 1 4 7', 'Val 5 8 10', 'Val 2 9 11', 'Val 3 6 12',
                  'Val 1 6 9', 'Val 2 7 10', 'Val 3 8 11', 'Val 4 5 12']

fig = go.Figure()
fig.add_trace(go.Scatter(
    x=validationSets,
    y=meanOfMeansC2_12,
    error_y=dict(
                type='data',
                symmetric=True,
                color='black',
                thickness=1,
                width=8,
                array=semOfMeansC2_12),
    name='C2',
    mode='markers',
    marker=dict(color="#00429d", size=8)
))
fig.add_trace(go.Scatter(
    x=validationSets,
    y=meanOfMeansHybrid_12,
    error_y=dict(
                type='data',
                symmetric=True,
                color='black',
                thickness=1,
                width=8,
                array=semOfMeansHybrid_12),
    name='Hybrid',
    mode='markers',
    marker=dict(color="#93003a", size=8)

))

title="Means & SEM - 13 Features - Binary Crystals Crystal Rating (C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024)"

fig.update_layout(xaxis_tickangle=-45,
                  plot_bgcolor="#fff",
                  title=title,
                  xaxis_title="Validation Set",
                  yaxis_title="Mean",
                font=dict(
                    family="Helvetica",
                    size=9,
                    color="Black"))
fig.update_yaxes(showgrid=True, gridwidth=0.5, gridcolor='LightPink', minor_griddash="dot")
fig.update_layout(title_pad_l=400,
                  title_pad_r=400,
                  title={
                  'text': title,
                  'y':0.9,
                  'x':0.5,
                  'xanchor': 'center',
                  'yanchor': 'top'})

fig.show()

fig.write_image(path + "/" + "Means and SEM - C2 vs Hybrid Networks.png")
```

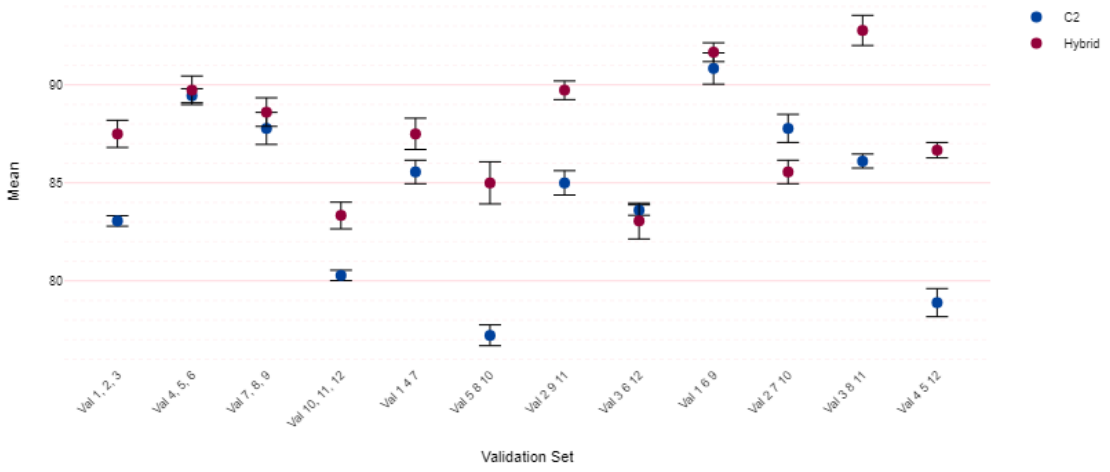


Figure 7.2: Means and Standard Error of the Mean (SEM) Between Validation Sets - Analysis and Visualisation coded in Jupyter Notebooks using Plotly for Visualisation

7.1.5 Comparing the Accuracy of Rock Predictions - Hybrid - Continuous vs Binary Crystal Ratings

In [1]:
```
import os
import pandas as pd
import numpy as np
import fnmatch

root = "C:/Users/c21012241/Dropbox"

##### Change paths as required
## 12 features - Binary crystals
path1 = root + "/12 Features - Binary Crystals + No Brightness/\
C2 LR 10^-3 E 200 MB 1024 - H LR 10^-3 E 50 MiniBatch 1024 - 12U LR 10^-3 E 200 MB 1024 - 12 of 12"

## 12 features - Continuous crystals
path2 = root + "/12 Features - Continuous Crystals + No Bright/\
C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024 - 12U LR10^-3 E200 MB1024 - 12of12"

# For use later
Feature_Analysis_Path_12 = root +"/Cardiff/Dissertation/XAI&I - Dissertation/\
Continuous vs Binary Crystals/12 Features - No Brightness/"

#Feature_Analysis_Path_13 = root + "/Cardiff/Dissertation/XAI&I - Dissertation/\
#Continuous vs Binary Crystals/13 Features"

model_parameter_name = "C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024"

filePath = Feature_Analysis_Path_12 + model_parameter_name + "/"
#filePath = Feature_Analysis_Path_13 + model_parameter_name + "/"


# Create lists of file paths
hybrid_binary_crystals_path1 = sorted([os.path.join(path1, file) for file in os.listdir(path1)], key=os.path.getmtime)
hybrid_continuous_crystals_path2 = sorted([os.path.join(path2, file) for file in os.listdir(path2)], key=os.path.getmtime)

# Rock names and the three variations of image arrangements used during training
rockNamesTen = ["Granite", "Obsidian", "Pegmatite", "Pumice", "Gneiss", "Marble", "Slate", "Breccia", "Conglomerate", "Sandstone"]
Ar1_index = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Ar2_index = [1, 4, 7, 5, 8, 10, 2, 9, 11, 3, 6, 12]
Ar3_index = [1, 6, 9, 2, 7, 10, 3, 8, 11, 4, 5, 12]

#pd.set_option('display.float_format', '{:.10f}'.format)
pd.set_option('display.float_format', '{:.2%}'.format)

# List with 30 rock names i.e. each rock has 3 validation images
all_rock_names = []
for name in rockNamesTen:
    all_rock_names.extend([name] * 3)

# Dictionary to store DataFrames
bi_Val_Hybrids = {key: pd.DataFrame() for key in ['1_2_3', '4_5_6', '7_8_9', '10_11_12', '1_4_7', '5_8_10',
                                                  '2_9_11', '3_6_12', '1_6_9', '2_7_10', '3_8_11', '4_5_12']}
con_Val_Hybrids = bi_Val_Hybrids.copy()

## Two keywords due to spelling mistakes or change of name
#keyword_Hybrid_1 = "netHybrid - Confidence of Prediciton"
#keyword_Hybrid_2 = "netHybrid - Confidence of Prediciton"

keyword_Hybrid_1 = "netHybrid-ConfidencePred"
keyword_Hybrid_2 = "netHybrid-ConfidencePred"

# Keywords of the 12 validation rock orders used
keyword_01_02_03 = "Val_1 2 3"
keyword_04_05_06 = "Val_4 5 6"
keyword_07_08_09 = "Val_7 8 9"
keyword_10_11_12 = "Val_10 11 12"
keyword_01_04_07 = "Val_1 4 7"
keyword_05_08_10 = "Val_5 8 10"
keyword_02_09_11 = "Val_2 9 11"
keyword_03_06_12 = "Val_3 6 12"
keyword_01_06_09 = "Val_1 6 9"
keyword_02_07_10 = "Val_2 7 10"
keyword_03_08_11 = "Val_3 8 11"
keyword_04_05_12 = "Val_4 5 12"
```

In [2]:
```
# A function to read the data from the csv files
def read_DataFrames(file_list, val_Keywords, data_Type_Keyword, all_rock_names):
    dfs = {}
    for file in file_list:
        if any(keyword in file for keyword in val_Keywords) and data_Type_Keyword in file:
            for keyword in val_Keywords:
                if keyword in file:
                    df_name = f"{keyword.replace('*', '').replace('?', '')}"
                    df = pd.read_csv(file, header=None)
                    df.columns = list(all_rock_names)
                    dfs[df_name] = df
                    break  # Break the loop after the first match
    return dfs
```

In [3]:

```
# Define keywords for arrangement 1, 2 and 3
val_Keywords_Ar1 = ["Val_1 2 3", "Val_4 5 6", "Val_7 8 9", "Val_10 11 12"]

val_Keywords_Ar2 = ["Val_1 4 7", "Val_5 8 10", "Val_2 9 11", "Val_3 6 12"]

val_Keywords_Ar3 = ["Val_1 6 9", "Val_2 7 10", "Val_3 8 11", "Val_4 5 12"]

# Ar1
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar1_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar1, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar1_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar1, keyword_Hybrid_2,all_rock_names)

#Ar2
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar2_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar2, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar2_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar2, keyword_Hybrid_2, all_rock_names)

#Ar3
# Read and assign DataFrames for hybrid_binary_crystals_path1
Ar3_bi_Val_Hybrid = read_DataFrames(hybrid_binary_crystals_path1, val_Keywords_Ar3, keyword_Hybrid_1, all_rock_names)
# Read and assign DataFrames for hybrid_continuous_crystals_path2
Ar3_con_Val_Hybrid = read_DataFrames(hybrid_continuous_crystals_path2, val_Keywords_Ar3, keyword_Hybrid_2, all_rock_names)
```

In [4]:

```
# Function to name the rocks in each set of three validation images
def arrangeValidationByRockName(dict_of_dfs):
    rock_orders = [
        ("Granite", 0, 3),
        ("Obsidian", 3, 6),
        ("Pegmatite", 6, 9),
        ("Pumice", 9, 12),
        ("Gneiss", 12, 15),
        ("Marble", 15, 18),
        ("Slate", 18, 21),
        ("Breccia", 21, 24),
        ("Conglomerate", 24, 27),
        ("Sandstone", 27, None)
    ]

    categorized_dfs = {rock_name: pd.DataFrame() for rock_name, _, _ in rock_orders}

    for df_name, df in dict_of_dfs.items():
        for rock_name, start, end in rock_orders:
            if end is None:
                categorized_dfs[rock_name] = pd.concat([categorized_dfs[rock_name], df.iloc[:, start:]], axis=1)
            else:
                categorized_dfs[rock_name] = pd.concat([categorized_dfs[rock_name], df.iloc[:, start:end]], axis=1)

    return categorized_dfs.values()
```

In [5]:

```
# Arrange by rock name (Ar1 already is?) and turn into a list
Ar1_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar1_bi_Val_Hybrid))
Ar1_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar1_con_Val_Hybrid))
Ar2_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar2_bi_Val_Hybrid))
Ar2_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar2_con_Val_Hybrid))
Ar3_bi_Val_Sorted_Name = list(arrangeValidationByRockName(Ar3_bi_Val_Hybrid))
Ar3_con_Val_Sorted_Name = list(arrangeValidationByRockName(Ar3_con_Val_Hybrid))

# Extracted DataFrames for each rock type and arrangement
Granite_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[0]
Obsidian_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[1]
Pegmatite_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[2]
Pumice_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[3]
Gneiss_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[4]
Marble_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[5]
Slate_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[6]
Breccia_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[7]
Conglomerate_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[8]
Sandstone_Ar1_bi_Val_Sorted_Name = Ar1_bi_Val_Sorted_Name[9]

Granite_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[0]
Obsidian_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[1]
Pegmatite_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[2]
Pumice_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[3]
Gneiss_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[4]
Marble_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[5]
Slate_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[6]
Breccia_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[7]
Conglomerate_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[8]
Sandstone_Ar1_con_Val_Sorted_Name = Ar1_con_Val_Sorted_Name[9]

Granite_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[0]
Obsidian_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[1]
Pegmatite_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[2]
Pumice_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[3]
Gneiss_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[4]
Marble_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[5]
Slate_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[6]
Breccia_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[7]
Conglomerate_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[8]
Sandstone_Ar2_bi_Val_Sorted_Name = Ar2_bi_Val_Sorted_Name[9]

Granite_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[0]
Obsidian_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[1]
Pegmatite_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[2]
Pumice_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[3]
Gneiss_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[4]
Marble_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[5]
Slate_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[6]
Breccia_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[7]
Conglomerate_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[8]
Sandstone_Ar2_con_Val_Sorted_Name = Ar2_con_Val_Sorted_Name[9]

Granite_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[0]
Obsidian_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[1]
Pegmatite_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[2]
Pumice_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[3]
Gneiss_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[4]
Marble_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[5]
Slate_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[6]
Breccia_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[7]
Conglomerate_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[8]
Sandstone_Ar3_bi_Val_Sorted_Name = Ar3_bi_Val_Sorted_Name[9]

Granite_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[0]
Obsidian_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[1]
Pegmatite_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[2]
Pumice_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[3]
Gneiss_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[4]
Marble_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[5]
Slate_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[6]
Breccia_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[7]
Conglomerate_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[8]
Sandstone_Ar3_con_Val_Sorted_Name = Ar3_con_Val_Sorted_Name[9]
```

In [6]:

```
# Function to rearrange data based on new index variables for arrangement 2 & 3
def setSortIndex(rockName, index):
    rockName = rockName.set_axis([index], axis = 1)
    rockName = rockName.sort_index(axis = 1)
    return rockName
```

In [7]:

```
# Lets use the function then :)

sorted_1to12_Granite_Ar1_bi_Val_Sorted_Name = setSortIndex(Granite_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Obsidian_Ar1_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pegmatite_Ar1_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pumice_Ar1_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Gneiss_Ar1_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Marble_Ar1_bi_Val_Sorted_Name = setSortIndex(Marble_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Slate_Ar1_bi_Val_Sorted_Name = setSortIndex(Slate_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Breccia_Ar1_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Conglomerate_Ar1_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar1_bi_Val_Sorted_Name, Ar1_index)
sorted_1to12_Sandstone_Ar1_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar1_bi_Val_Sorted_Name, Ar1_index)

sorted_1to12_Granite_Ar1_con_Val_Sorted_Name = setSortIndex(Granite_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Obsidian_Ar1_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pegmatite_Ar1_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Pumice_Ar1_con_Val_Sorted_Name = setSortIndex(Pumice_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Gneiss_Ar1_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Marble_Ar1_con_Val_Sorted_Name = setSortIndex(Marble_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Slate_Ar1_con_Val_Sorted_Name = setSortIndex(Slate_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Breccia_Ar1_con_Val_Sorted_Name = setSortIndex(Breccia_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Conglomerate_Ar1_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar1_con_Val_Sorted_Name, Ar1_index)
sorted_1to12_Sandstone_Ar1_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar1_con_Val_Sorted_Name, Ar1_index)

sorted_1to12_Granite_Ar2_bi_Val_Sorted_Name = setSortIndex(Granite_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Obsidian_Ar2_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pegmatite_Ar2_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pumice_Ar2_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Gneiss_Ar2_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Marble_Ar2_bi_Val_Sorted_Name = setSortIndex(Marble_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Slate_Ar2_bi_Val_Sorted_Name = setSortIndex(Slate_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Breccia_Ar2_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Conglomerate_Ar2_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar2_bi_Val_Sorted_Name, Ar2_index)
sorted_1to12_Sandstone_Ar2_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar2_bi_Val_Sorted_Name, Ar2_index)

sorted_1to12_Granite_Ar2_con_Val_Sorted_Name = setSortIndex(Granite_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Obsidian_Ar2_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pegmatite_Ar2_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Pumice_Ar2_con_Val_Sorted_Name = setSortIndex(Pumice_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Gneiss_Ar2_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Marble_Ar2_con_Val_Sorted_Name = setSortIndex(Marble_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Slate_Ar2_con_Val_Sorted_Name = setSortIndex(Slate_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Breccia_Ar2_con_Val_Sorted_Name = setSortIndex(Breccia_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Conglomerate_Ar2_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar2_con_Val_Sorted_Name, Ar2_index)
sorted_1to12_Sandstone_Ar2_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar2_con_Val_Sorted_Name, Ar2_index)

sorted_1to12_Granite_Ar3_bi_Val_Sorted_Name = setSortIndex(Granite_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Obsidian_Ar3_bi_Val_Sorted_Name = setSortIndex(Obsidian_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pegmatite_Ar3_bi_Val_Sorted_Name = setSortIndex(Pegmatite_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pumice_Ar3_bi_Val_Sorted_Name = setSortIndex(Pumice_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Gneiss_Ar3_bi_Val_Sorted_Name = setSortIndex(Gneiss_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Marble_Ar3_bi_Val_Sorted_Name = setSortIndex(Marble_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Slate_Ar3_bi_Val_Sorted_Name = setSortIndex(Slate_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Breccia_Ar3_bi_Val_Sorted_Name = setSortIndex(Breccia_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Conglomerate_Ar3_bi_Val_Sorted_Name = setSortIndex(Conglomerate_Ar3_bi_Val_Sorted_Name, Ar3_index)
sorted_1to12_Sandstone_Ar3_bi_Val_Sorted_Name = setSortIndex(Sandstone_Ar3_bi_Val_Sorted_Name, Ar3_index)

sorted_1to12_Granite_Ar3_con_Val_Sorted_Name = setSortIndex(Granite_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Obsidian_Ar3_con_Val_Sorted_Name = setSortIndex(Obsidian_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pegmatite_Ar3_con_Val_Sorted_Name = setSortIndex(Pegmatite_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Pumice_Ar3_con_Val_Sorted_Name = setSortIndex(Pumice_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Gneiss_Ar3_con_Val_Sorted_Name = setSortIndex(Gneiss_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Marble_Ar3_con_Val_Sorted_Name = setSortIndex(Marble_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Slate_Ar3_con_Val_Sorted_Name = setSortIndex(Slate_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Breccia_Ar3_con_Val_Sorted_Name = setSortIndex(Breccia_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Conglomerate_Ar3_con_Val_Sorted_Name = setSortIndex(Conglomerate_Ar3_con_Val_Sorted_Name, Ar3_index)
sorted_1to12_Sandstone_Ar3_con_Val_Sorted_Name = setSortIndex(Sandstone_Ar3_con_Val_Sorted_Name, Ar3_index)
```

In [8]:

```
# Does what it says on the tin

def meanRockArrangements(Ar1, Ar2, Ar3):
    """Returns the mean of the three rock arrangements"""
    return (Ar1 + Ar2 + Ar3) / 3
```

In [9]:

```
# Dictionary of binary and continuous sets

arrangements_bi = ["Ar1_bi_Val_Sorted_Name", "Ar2_bi_Val_Sorted_Name", "Ar3_bi_Val_Sorted_Name"]

mean_accuracy_results_bi = {}

for rock_name in rockNamesTen:
    accuracy_list = []
    for arrangement in arrangements_bi:
        dataFrame_name = f"sorted_1to12_{rock_name}_{arrangement}"
        accuracy_list.append(eval(dataFrame_name))

    mean_accuracy = meanRockArrangements(*accuracy_list)
    mean_accuracy_results_bi[rock_name] = mean_accuracy

arrangements_con = ["Ar1_con_Val_Sorted_Name", "Ar2_con_Val_Sorted_Name", "Ar3_con_Val_Sorted_Name"]

mean_accuracy_results_con = {}

for rock_name in rockNamesTen:
    accuracy_list = []
    for arrangement in arrangements_con:
        dataFrame_name = f"sorted_1to12_{rock_name}_{arrangement}"
        accuracy_list.append(eval(dataFrame_name))

    mean_accuracy = meanRockArrangements(*accuracy_list)
    mean_accuracy_results_con[rock_name] = mean_accuracy
```

In [10]:

```
## Binary crystal rating by rock name

mean_accuracy_results_bi_key = next(iter(mean_accuracy_results_bi))

# Granite mean confidence
Granite_bi_mean_accuracy_results_key = list(mean_accuracy_results_bi.keys())[0]
Granite_bi_mean_accuracy_results = mean_accuracy_results_bi[Granite_bi_mean_accuracy_results_key].rename(index=dict(enumerate(rockNamesTen)))
# Obsidian mean confidence
Obsidian_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[1]
Obsidian_bi_mean_accuracy_results = mean_accuracy_results_bi[Obsidian_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Pegmatite mean confidence
Pegmatite_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[2]
Pegmatite_bi_mean_accuracy_results = mean_accuracy_results_bi[Pegmatite_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Pumice mean confidence
Pumice_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[3]
Pumice_bi_mean_accuracy_results = mean_accuracy_results_bi[Pumice_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Gneiss mean confidence
Gneiss_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[4]
Gneiss_bi_mean_accuracy_results = mean_accuracy_results_bi[Gneiss_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Marble mean confidence
Marble_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[5]
Marble_bi_mean_accuracy_results = mean_accuracy_results_bi[Marble_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Slate mean confidence
Slate_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[6]
Slate_bi_mean_accuracy_results = mean_accuracy_results_bi[Slate_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Breccia mean confidence
Breccia_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[7]
Breccia_bi_mean_accuracy_results = mean_accuracy_results_bi[Breccia_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Conglomerate mean confidence
Conglomerate_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[8]
Conglomerate_bi_mean_accuracy_results = mean_accuracy_results_bi[Conglomerate_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))
# Sandstone mean confidence
Sandstone_bi_mean_accuracy_results_bi_key = list(mean_accuracy_results_bi.keys())[9]
Sandstone_bi_mean_accuracy_results = mean_accuracy_results_bi[Sandstone_bi_mean_accuracy_results_bi_key].rename(index=dict(enumerate(rockNamesTen)))

## Continuous crystal rating by rock name

mean_accuracy_results_con_key = next(iter(mean_accuracy_results_con))

# Granite mean confidence
Granite_con_mean_accuracy_results_key = list(mean_accuracy_results_con.keys())[0]
Granite_con_mean_accuracy_results = mean_accuracy_results_con[Granite_con_mean_accuracy_results_key].rename(index=dict(enumerate(rockNamesTen)))
# Obsidian mean confidence
Obsidian_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[1]
Obsidian_con_mean_accuracy_results = mean_accuracy_results_con[Obsidian_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Pegmatite mean confidence
Pegmatite_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[2]
Pegmatite_con_mean_accuracy_results = mean_accuracy_results_con[Pegmatite_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Pumice mean confidence
Pumice_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[3]
Pumice_con_mean_accuracy_results = mean_accuracy_results_con[Pumice_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Gneiss mean confidence
Gneiss_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[4]
Gneiss_con_mean_accuracy_results = mean_accuracy_results_con[Gneiss_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Marble mean confidence
Marble_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[5]
Marble_con_mean_accuracy_results = mean_accuracy_results_con[Marble_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Slate mean confidence
Slate_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[6]
Slate_con_mean_accuracy_results = mean_accuracy_results_con[Slate_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Breccia mean confidence
Breccia_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[7]
Breccia_con_mean_accuracy_results = mean_accuracy_results_con[Breccia_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Conglomerate mean confidence
Conglomerate_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[8]
Conglomerate_con_mean_accuracy_results = mean_accuracy_results_con[Conglomerate_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
# Sandstone mean confidence
Sandstone_con_mean_accuracy_results_con_key = list(mean_accuracy_results_con.keys())[9]
Sandstone_con_mean_accuracy_results = mean_accuracy_results_con[Sandstone_con_mean_accuracy_results_con_key].rename(index=dict(enumerate(rockNamesTen)))
```

In [11]:
```
Feature_Analysis_Path_12 = root + "/Cardiff/Dissertation/XAI&I - Dissertation/Continuous vs Binary Crystals/12 Features - No Brightness/"
#Feature_Analysis_Path_13 = "C:/Users/c21012241/Dropbox/Cardiff/Dissertation/XAI&I - Dissertation/Continuous vs Binary Crystals/13 Features"

model_parameter_name = "C2 LR10^-3 E200 MB1024 - H LR10^-3 E50 MB1024"

filePath = Feature_Analysis_Path_12 + model_parameter_name + "/"
#filePath = Feature_Analysis_Path_13 + model_parameter_name + "/"

Granite_Comparison = Granite_con_mean_accuracy_results.compare(Granite_bi_mean_accuracy_results)
Granite_Comparison = Granite_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Granite_Comparison.to_csv(filePath + "Granite_Comparison" + ".csv")

Obsidian_Comparison = Obsidian_con_mean_accuracy_results.compare(Obsidian_bi_mean_accuracy_results)
Obsidian_Comparison = Obsidian_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Obsidian_Comparison.to_csv(filePath + "Obsidian_Comparison" + ".csv")

Pegmatite_Comparison = Pegmatite_con_mean_accuracy_results.compare(Pegmatite_bi_mean_accuracy_results)
Pegmatite_Comparison = Pegmatite_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Pegmatite_Comparison.to_csv(filePath + "Pegmatite_Comparison" + ".csv")

Pumice_Comparison = Pumice_con_mean_accuracy_results.compare(Pumice_bi_mean_accuracy_results)
Pumice_Comparison = Pumice_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Pumice_Comparison.to_csv(filePath + "Pumice_Comparison" + ".csv")

Gneiss_Comparison = Gneiss_con_mean_accuracy_results.compare(Gneiss_bi_mean_accuracy_results)
Gneiss_Comparison = Gneiss_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Gneiss_Comparison.to_csv(filePath + "Gneiss_Comparison" + ".csv")

Marble_Comparison = Marble_con_mean_accuracy_results.compare(Marble_bi_mean_accuracy_results)
Marble_Comparison = Marble_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Marble_Comparison.to_csv(filePath + "Marble_Comparison" + ".csv")

Slate_Comparison = Slate_con_mean_accuracy_results.compare(Slate_bi_mean_accuracy_results)
Slate_Comparison = Slate_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Slate_Comparison.to_csv(filePath + "Slate_Comparison" + ".csv")

Breccia_Comparison = Breccia_con_mean_accuracy_results.compare(Breccia_bi_mean_accuracy_results)
Breccia_Comparison = Breccia_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Breccia_Comparison.to_csv(filePath + "Breccia_Comparison" + ".csv")

Conglomerate_Comparison = Conglomerate_con_mean_accuracy_results.compare(Conglomerate_bi_mean_accuracy_results)
Conglomerate_Comparison = Conglomerate_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Conglomerate_Comparison.to_csv(filePath + "Conglomerate_Comparison" + ".csv")

Sandstone_Comparison = Sandstone_con_mean_accuracy_results.compare(Sandstone_bi_mean_accuracy_results)
Sandstone_Comparison = Sandstone_Comparison.rename(columns={"self": "Continuous", "other": "Binary"})
Sandstone_Comparison.to_csv(filePath + "Sandstone_Comparison" + ".csv")
```

In [12]:
```
print(Granite_Comparison)
```

```
                  1                2                3                4   \
         Continuous Binary Continuous Binary Continuous Binary Continuous
Granite      99.96% 99.95%     99.82% 99.49%     91.86% 93.96%     99.48%
Obsidian      0.00%  0.00%      0.00%  0.00%      0.07%  0.02%      0.00%
Pegmatite     0.03%  0.05%      0.03%  0.02%      0.86%  0.53%      0.01%
Pumice        0.00%  0.00%      0.00%  0.00%      0.00%  0.00%      0.00%
Gneiss        0.00%  0.00%      0.03%  0.09%      5.66%  4.54%      0.51%
Marble        0.00%  0.00%      0.11%  0.39%      0.03%  0.27%      0.00%
Slate         0.00%  0.00%      0.00%  0.00%      0.00%  0.01%      0.00%
Breccia       0.00%  0.00%      0.00%  0.00%      0.60%  0.26%      0.00%
Conglomerate  0.00%  0.00%      0.00%  0.00%      0.92%  0.41%      0.00%
Sandstone     0.00%  0.00%      0.00%  0.00%      0.00%  0.00%      0.00%

                  5          ...         8                9   \
         Binary Continuous Binary  ... Continuous Binary Continuous
Granite      98.28%     84.00% 91.13%  ...     66.05% 60.80%     89.49%
Obsidian      0.00%      0.01%  0.03%  ...      0.01%  0.10%      0.04%
Pegmatite     0.00%      8.75%  3.77%  ...     14.66% 19.95%      3.00%
Pumice        0.00%      0.00%  0.02%  ...      0.00%  0.00%      0.00%
Gneiss        1.71%      5.93%  4.52%  ...     11.12% 12.73%      0.01%
Marble        0.00%      0.11%  0.02%  ...      0.05%  1.36%      2.00%
Slate         0.00%      0.00%  0.01%  ...      0.00%  0.00%      0.00%
Breccia       0.00%      0.30%  0.14%  ...      7.21%  3.82%      2.50%
Conglomerate  0.00%      0.87%  0.34%  ...      0.89%  1.23%      2.96%
Sandstone     0.00%      0.03%  0.01%  ...      0.00%  0.01%      0.00%

                 10               11               12
         Binary Continuous Binary Continuous Binary Continuous Binary
Granite      83.42%     94.81% 90.78%     70.77% 78.28%     77.48% 66.93%
Obsidian      0.54%      0.00%  0.00%      0.01%  0.02%      0.07%  0.10%
Pegmatite     9.00%      5.19%  9.22%     27.91% 19.20%      9.25% 17.80%
Pumice        0.00%      0.00%  0.00%      0.00%  0.00%      0.00%  0.00%
Gneiss        0.00%      0.00%  0.00%      0.12%  0.19%      0.12%  0.02%
Marble        5.34%      0.00%  0.00%      1.10%  1.86%      3.63%  5.31%
Slate         0.00%      0.00%  0.00%      0.00%  0.00%      0.02%  0.05%
Breccia       0.36%      0.00%  0.00%      0.01%  0.02%      6.95%  7.14%
Conglomerate  1.33%      0.00%  0.00%      0.08%  0.42%      2.48%  2.64%
Sandstone     0.00%      0.00%  0.00%      0.00%  0.00%      0.00%  0.01%

[10 rows x 24 columns]
```

Figure 7.3: Comparing the Accuracy of Rock Predictions - Hybrid - Continuous vs Binary Crystal Ratings

7.2 Data Visualisations

7.2.1 13 Features - Binary Crystal Rating

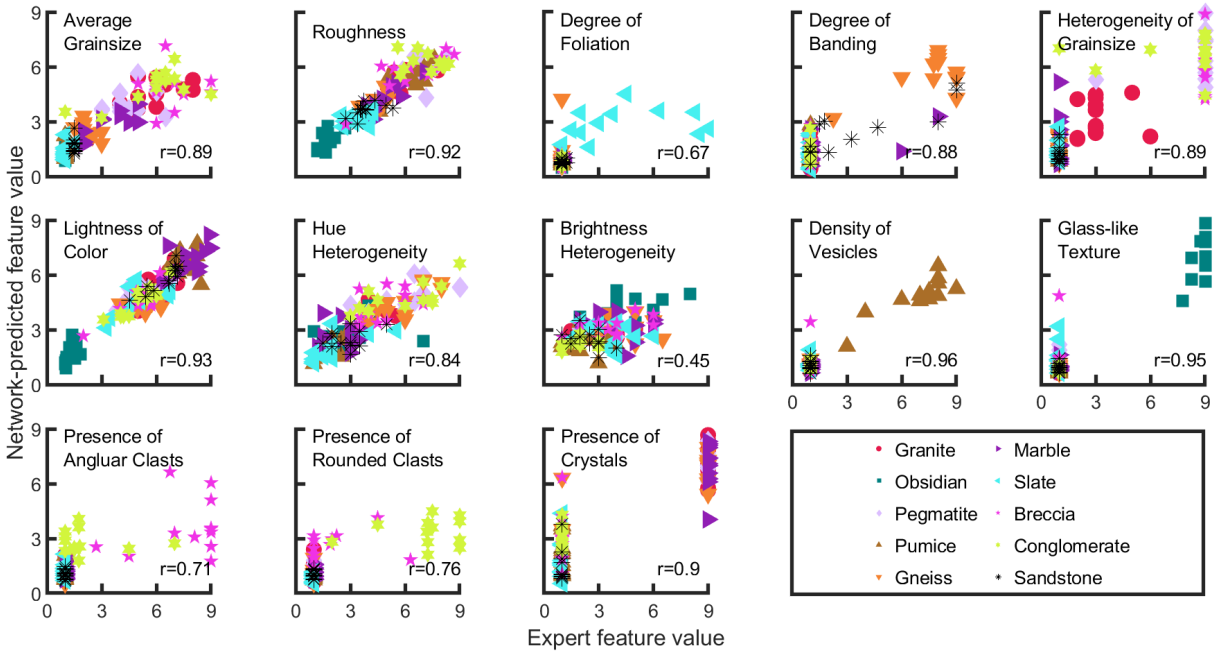


Figure 7.4: Feature Correlation - Sequential CBM Vs Expert Ratings - 13 Features - Binary Crystal Rating

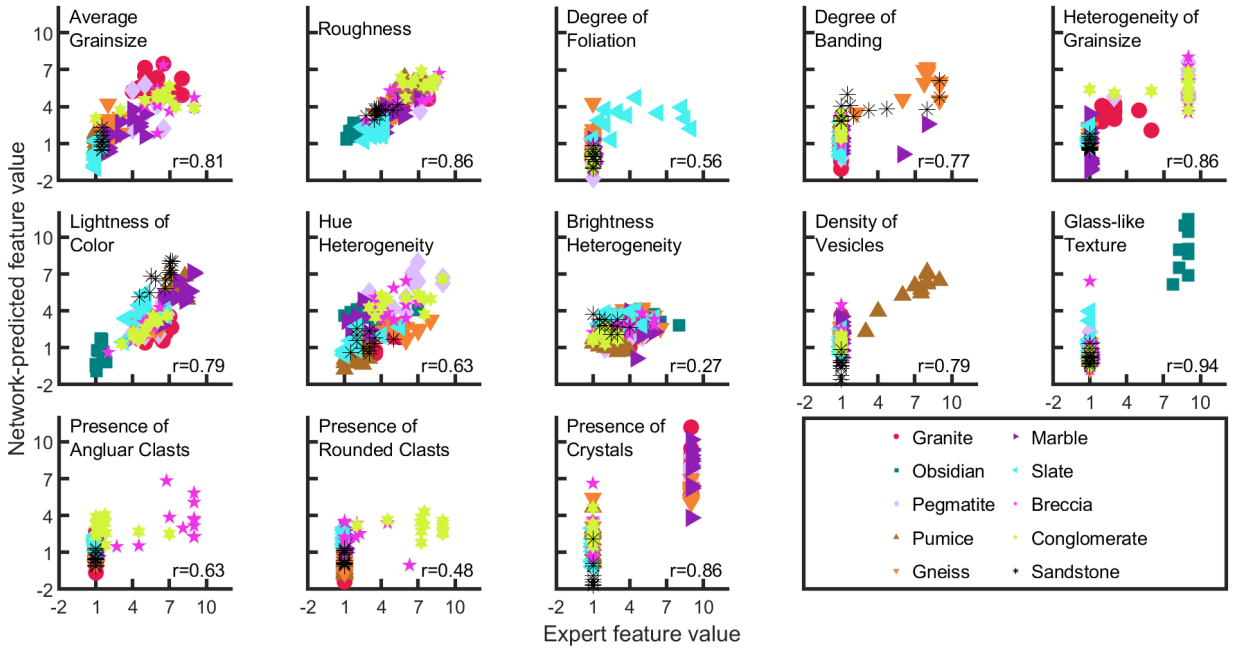


Figure 7.5: Feature Correlation - Hybrid Sequential CBM Vs Expert Ratings - 13 Features - Binary Crystal Rating

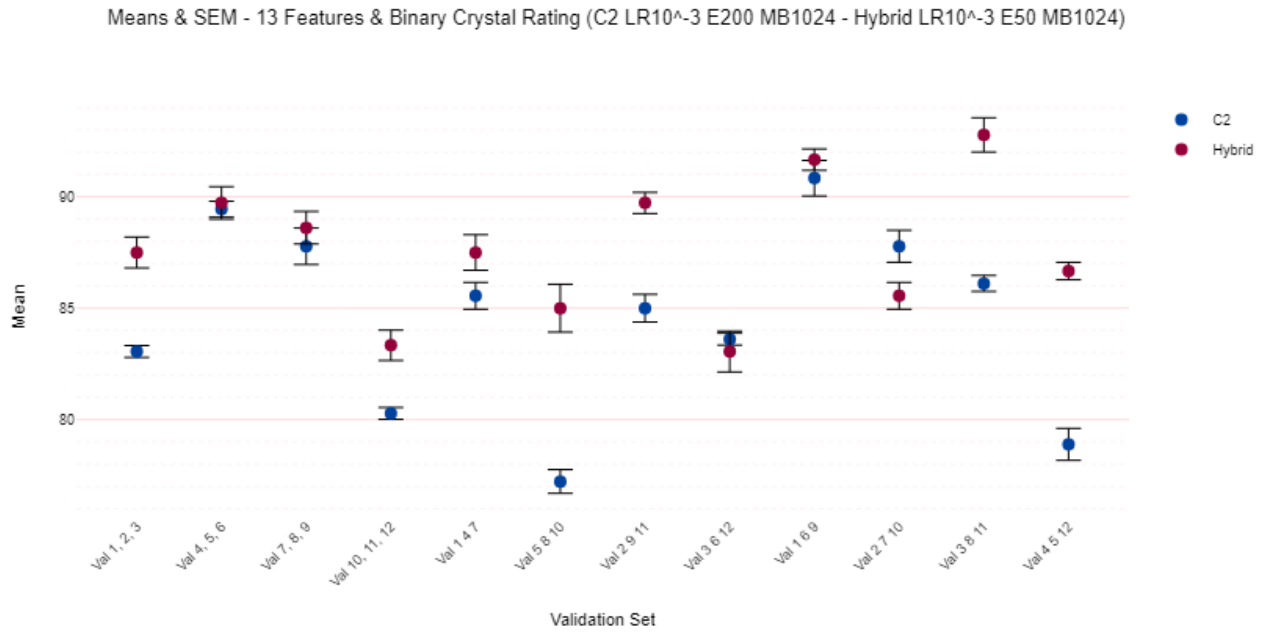


Figure 7.6: Means & SEM - Validation Sets - 13 Features - Binary Crystal Rating

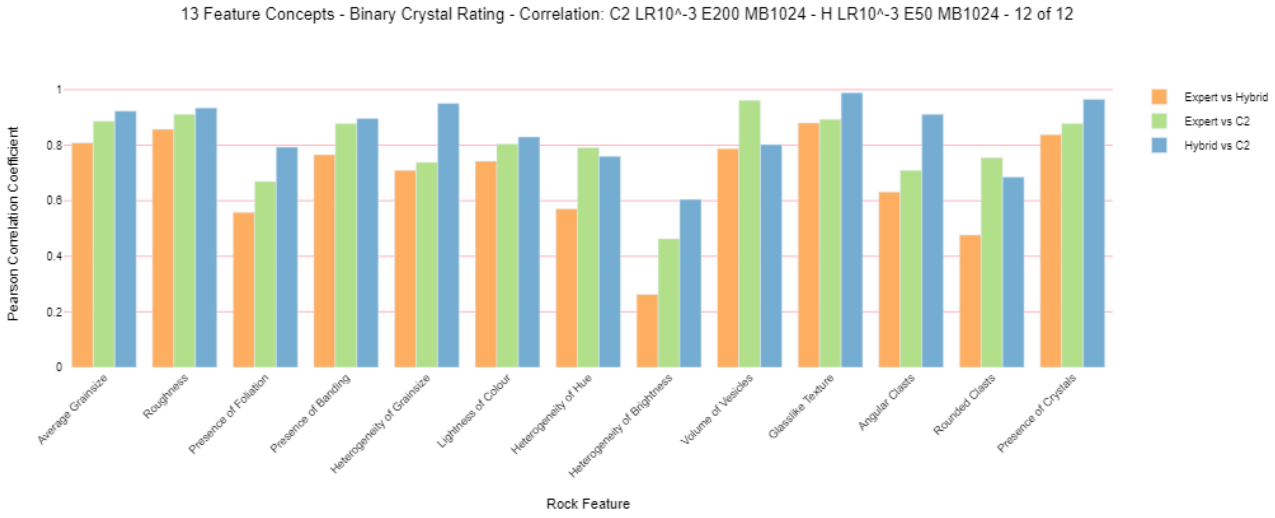


Figure 7.7: 13 Feature Concepts - Binary Crystal Rating - Mean correlation of Feature Concepts

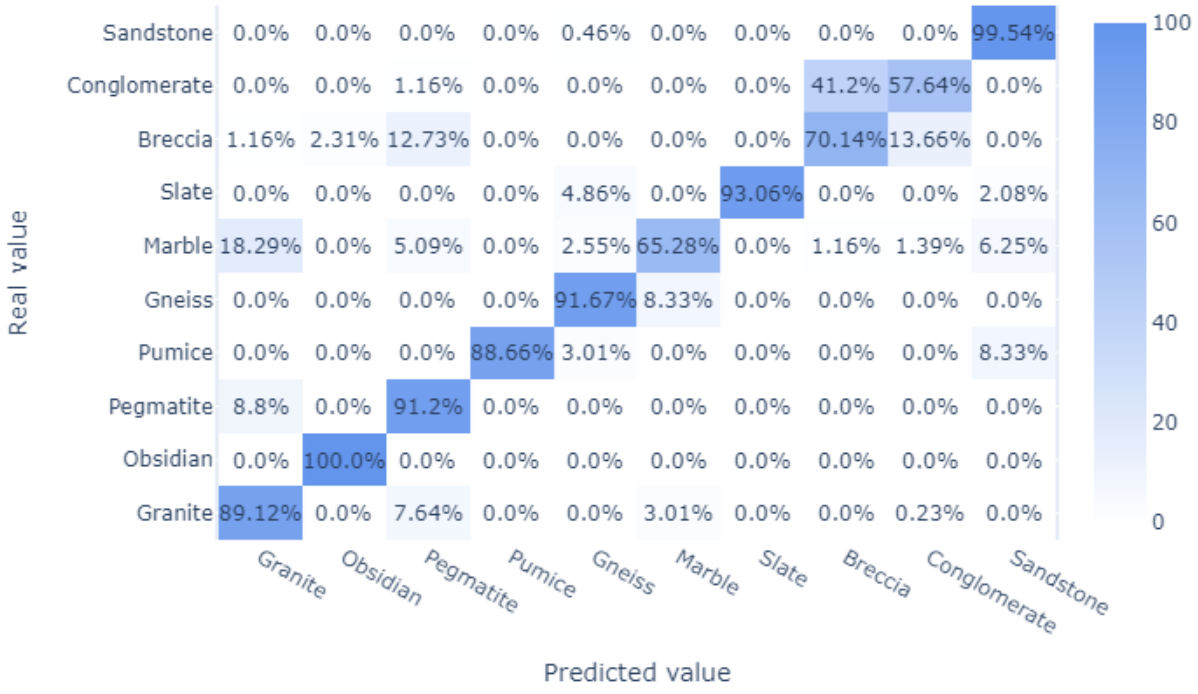


Figure 7.8: Sequential CBM Network Accuracy Confusion Matrix - 13 Features - Binary Crystal Ratings

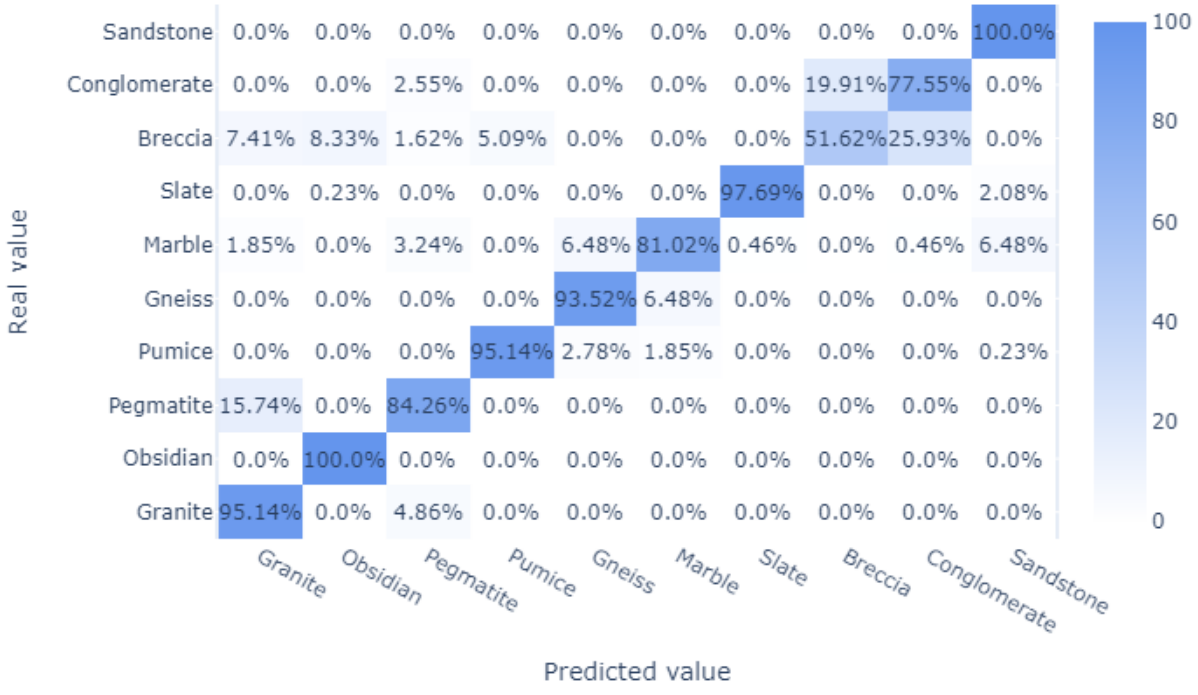


Figure 7.9: Hybrid Network Accuracy Confusion Matrix - 13 Features - Binary Crystal Ratings

7.2.2 13 Features - Continuous Crystal Rating

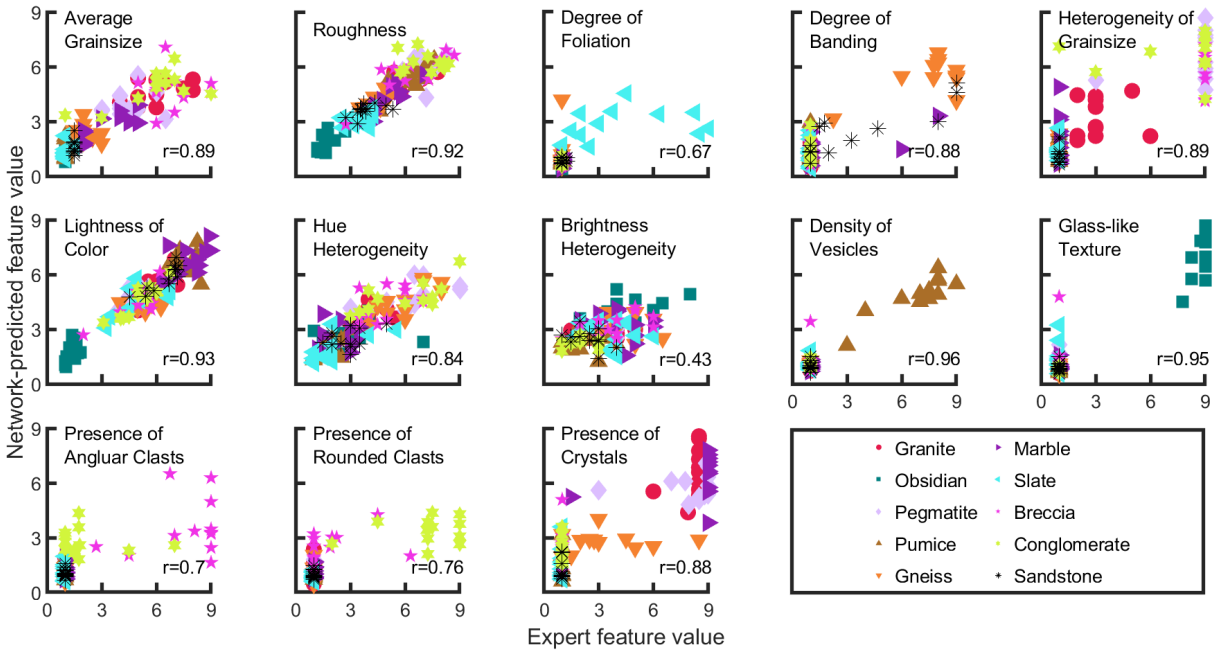


Figure 7.10: Feature Correlation - Sequential CBM Vs Expert Ratings - 13 Features - Continuous Crystal Rating

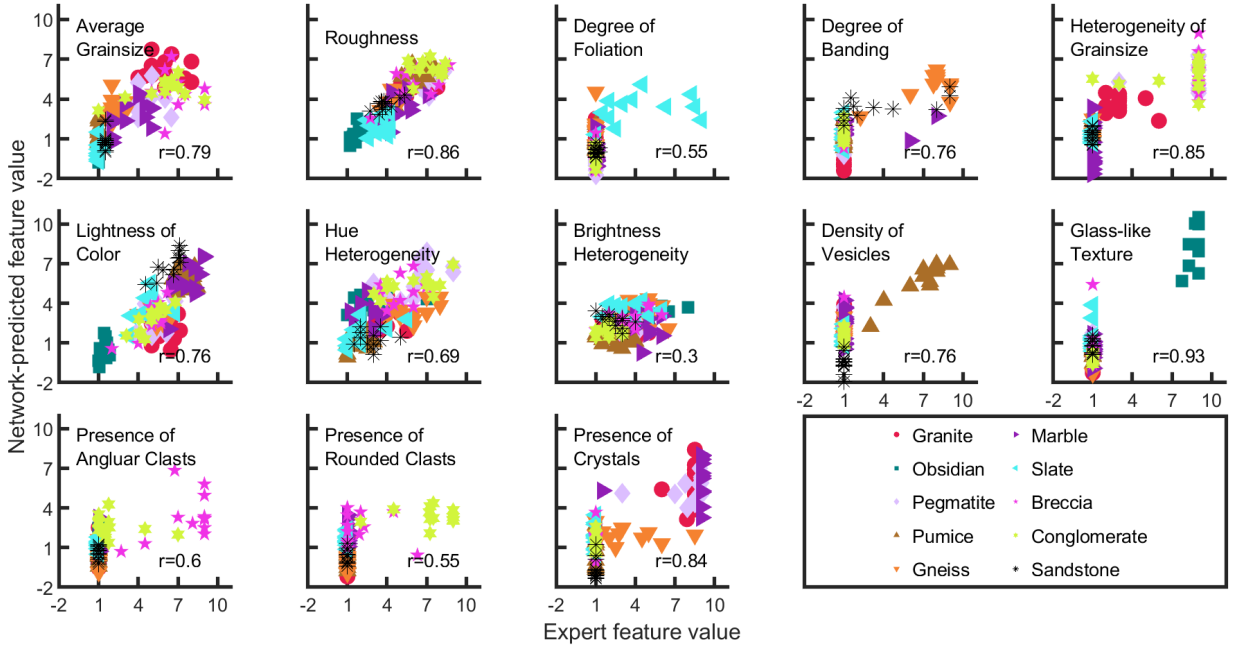


Figure 7.11: Feature Correlation - Hybrid Sequential CBM Vs Expert Ratings - 13 Features - Continuous Crystal Rating

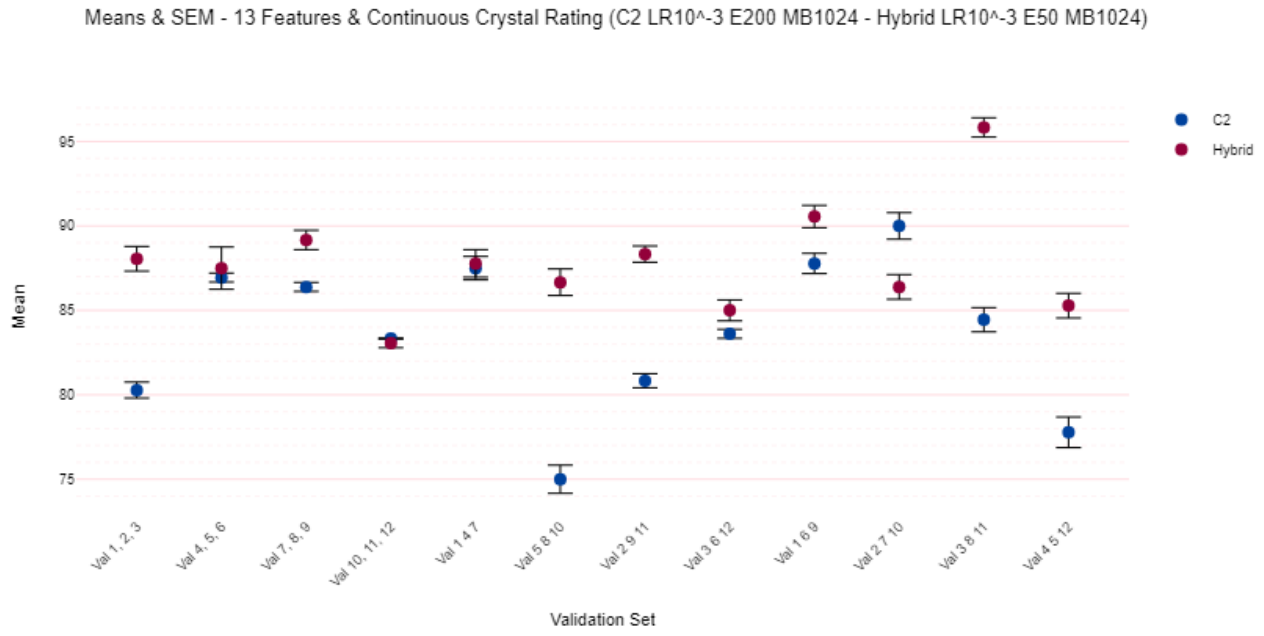


Figure 7.12: Means & SEM - Validation Sets - 13 Features - Continuous Crystal Rating

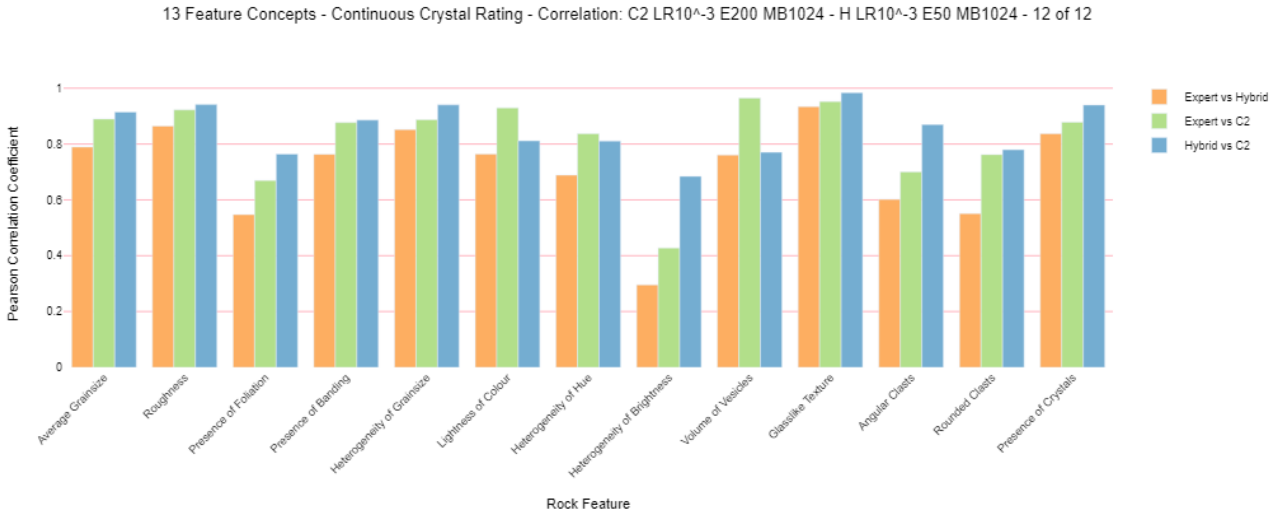


Figure 7.13: 13 Feature Concepts - Continuous Crystal Rating - Mean correlation of Feature Concepts

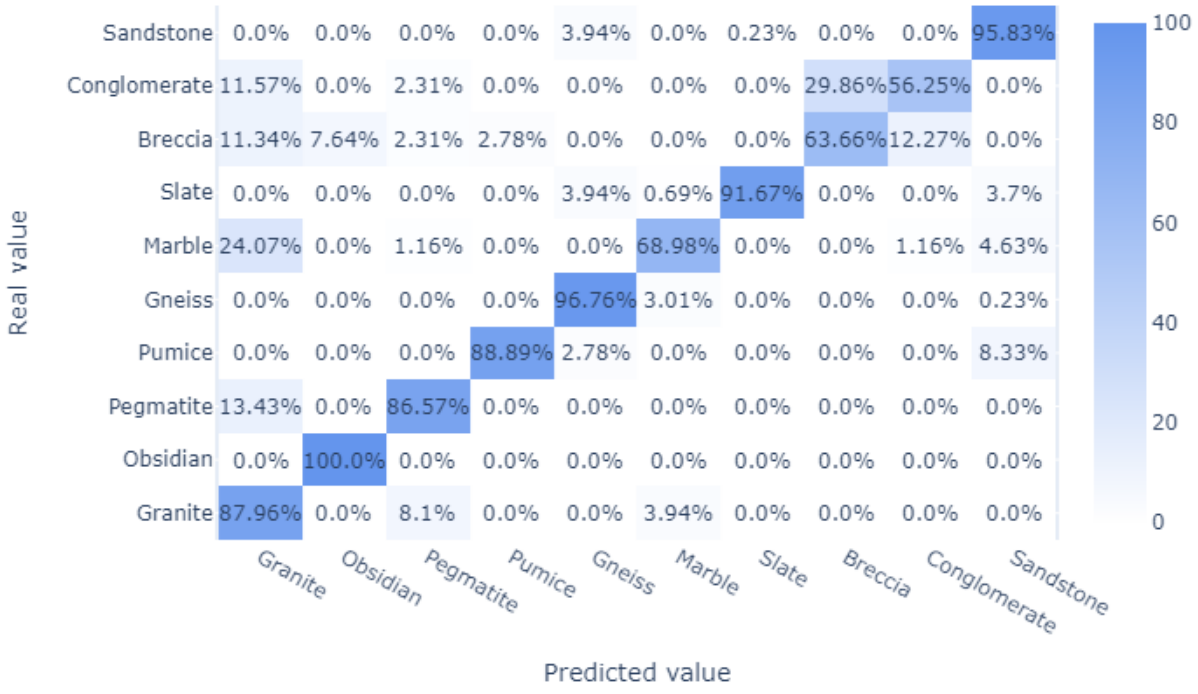


Figure 7.14: Sequential CBM Network Accuracy Confusion Matrix - 13 Features - Continuous Crystal Ratings

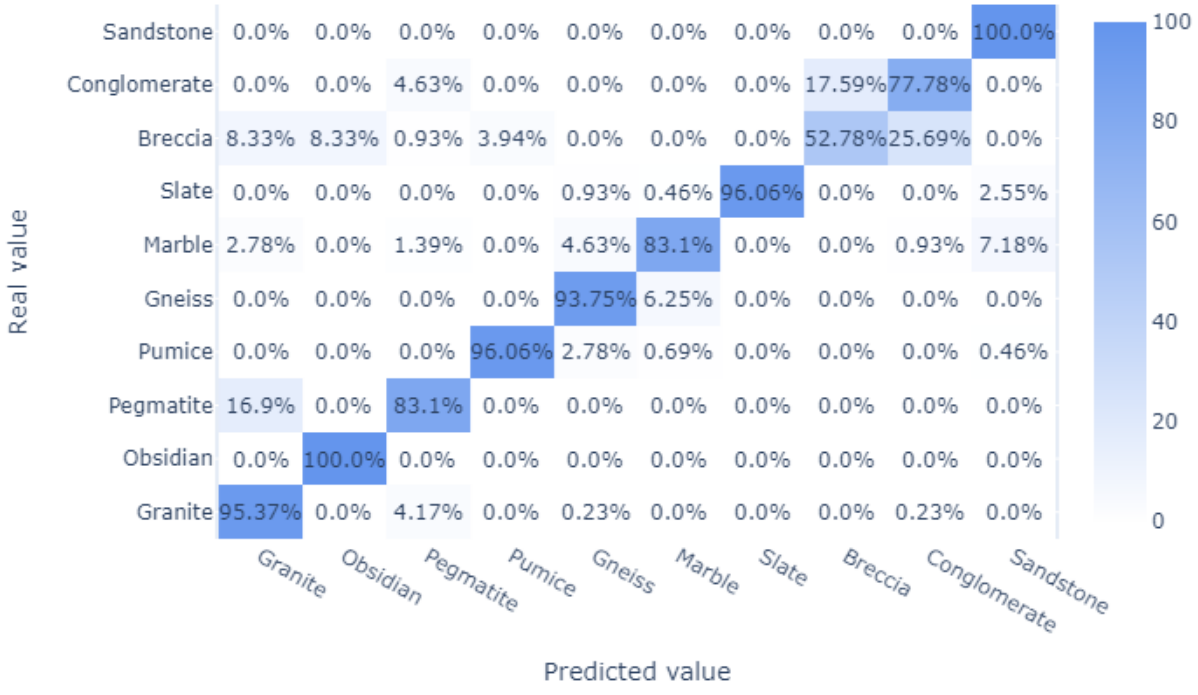


Figure 7.15: Hybrid Network Accuracy Confusion Matrix- 13 Features - Continuous Crystal Ratings

7.2.3 12 Features ("Heterogeneity of Brightness" Feature Removed) - Binary Crystal Rating

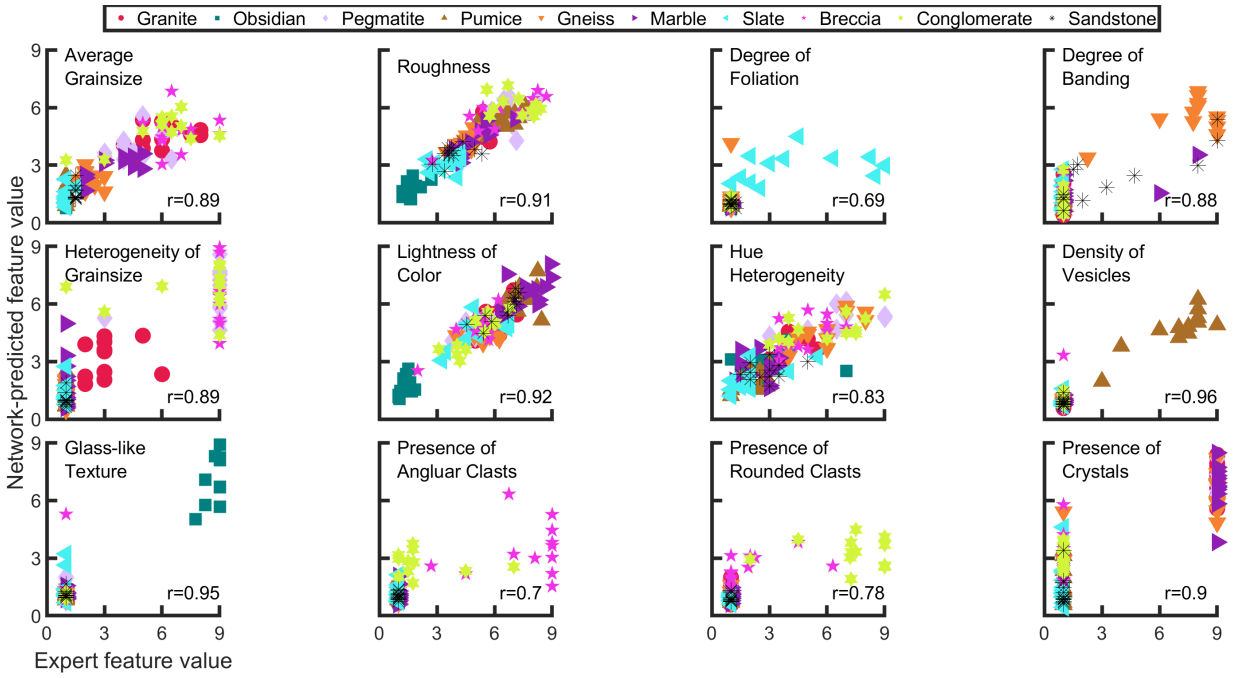


Figure 7.16: Feature Correlation - Sequential CBM Vs Expert Ratings - 12 Features - Binary Crystal Rating

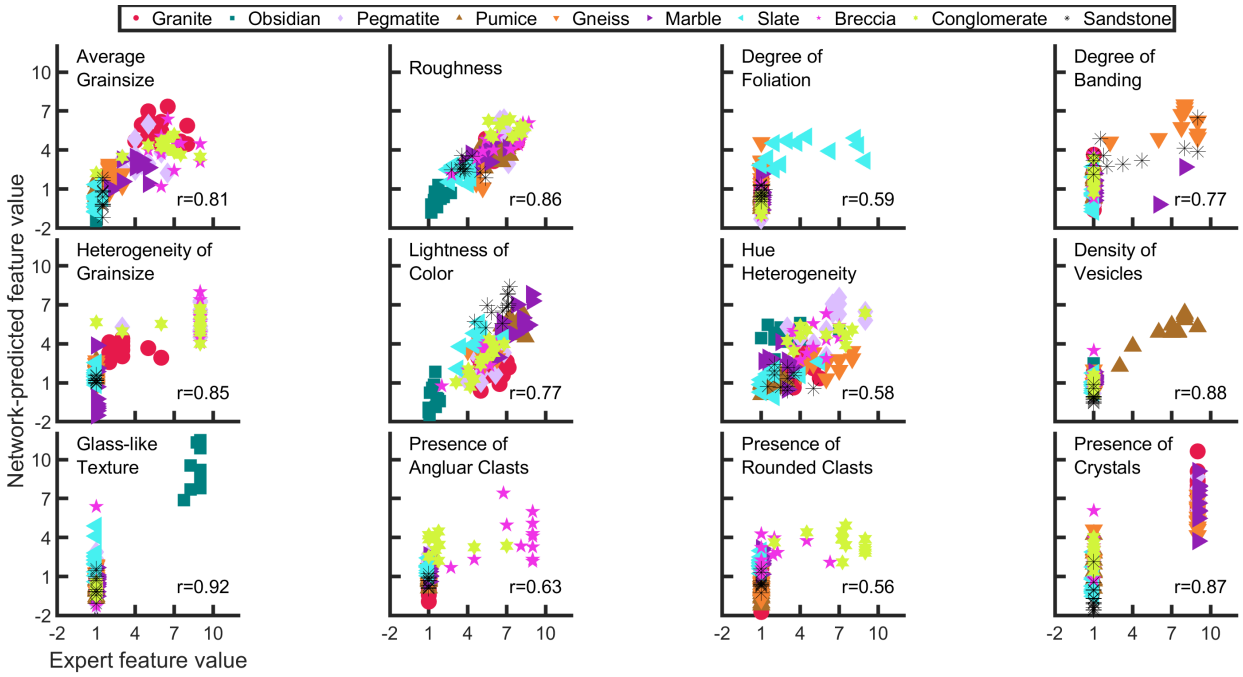


Figure 7.17: Feature Correlation - Hybrid Sequential CBM Vs Expert Ratings - 12 Features - Binary Crystal Rating

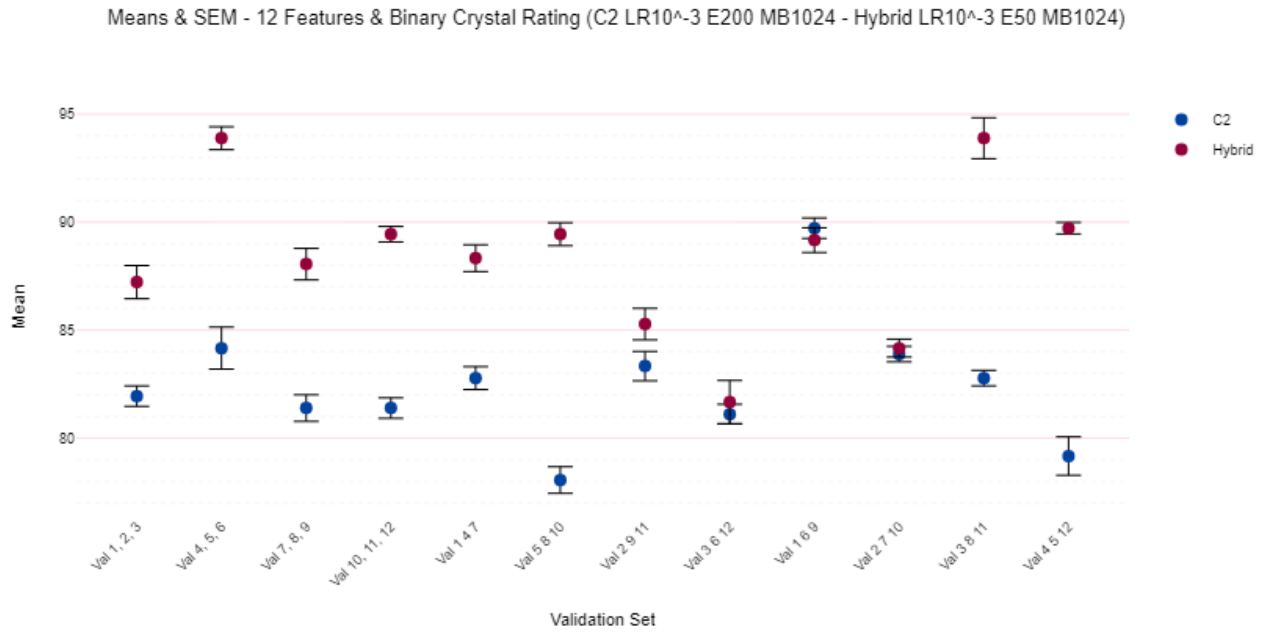


Figure 7.18: Means & SEM - Validation Sets - 12 Features - Binary Crystal Rating

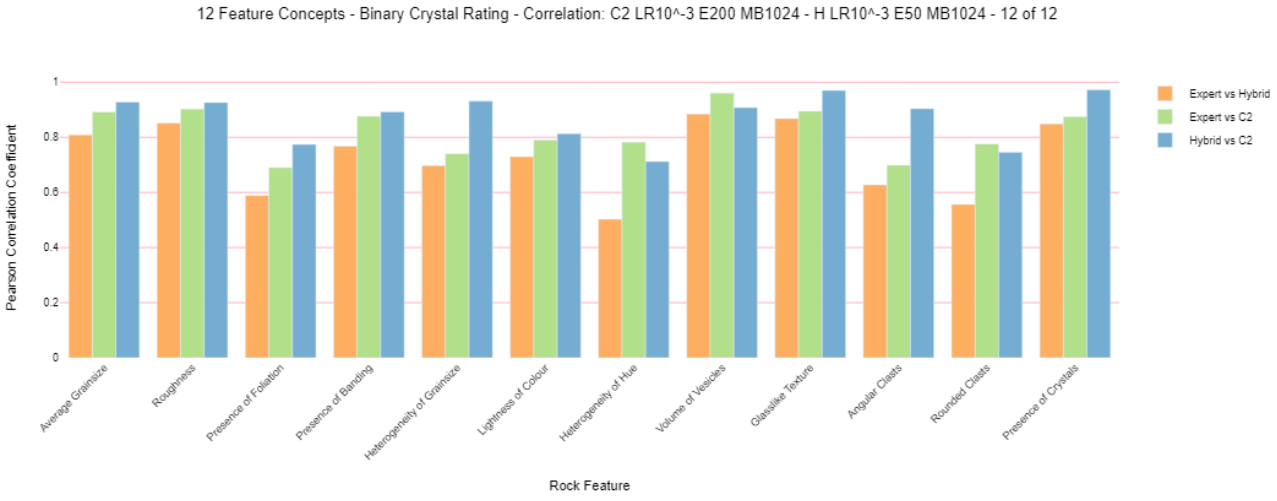


Figure 7.19: 12 Feature Concepts - Binary Crystal Rating - Mean correlation of Feature Concepts

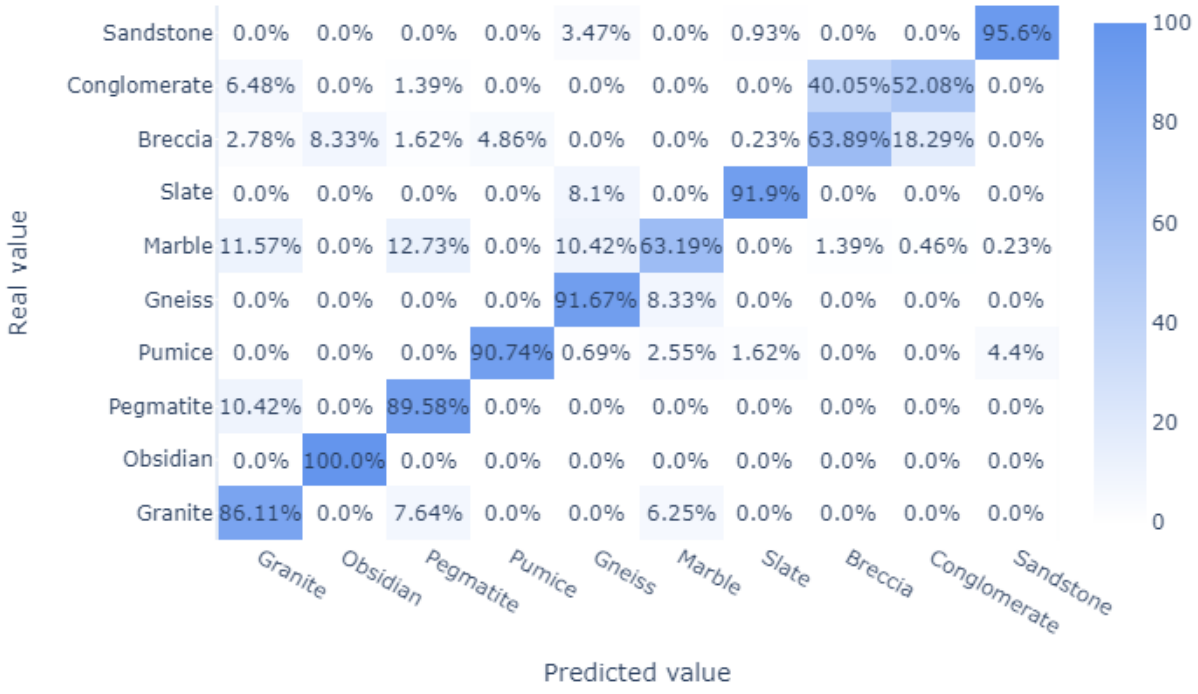


Figure 7.20: Sequential CBM Network Accuracy Confusion Matrix - 12 Features - Binary Crystal Ratings

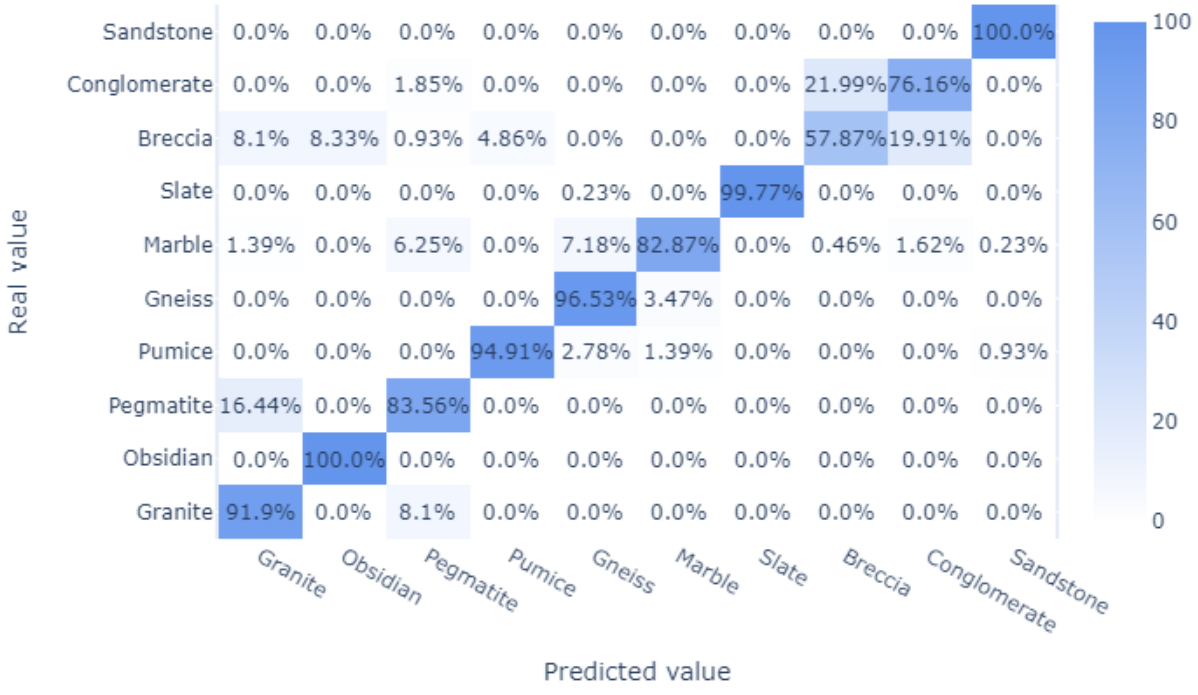


Figure 7.21: Hybrid Network Accuracy Confusion Matrix - 12 Features - Binary Crystal Ratings

7.2.4 12 Features ("Heterogeneity of Brightness" Feature Removed) - Continuous Crystal Rating

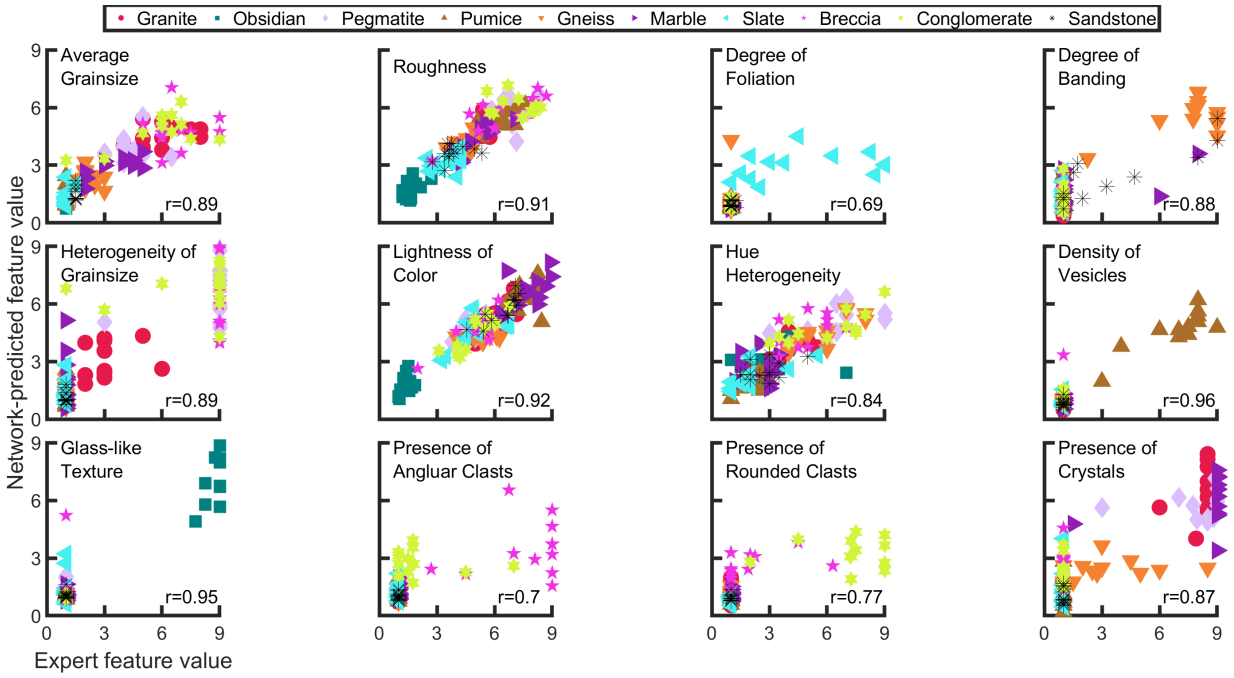


Figure 7.22: Feature Correlation - Sequential CBM Vs Expert Ratings - 12 Features - Continuous Crystal Rating

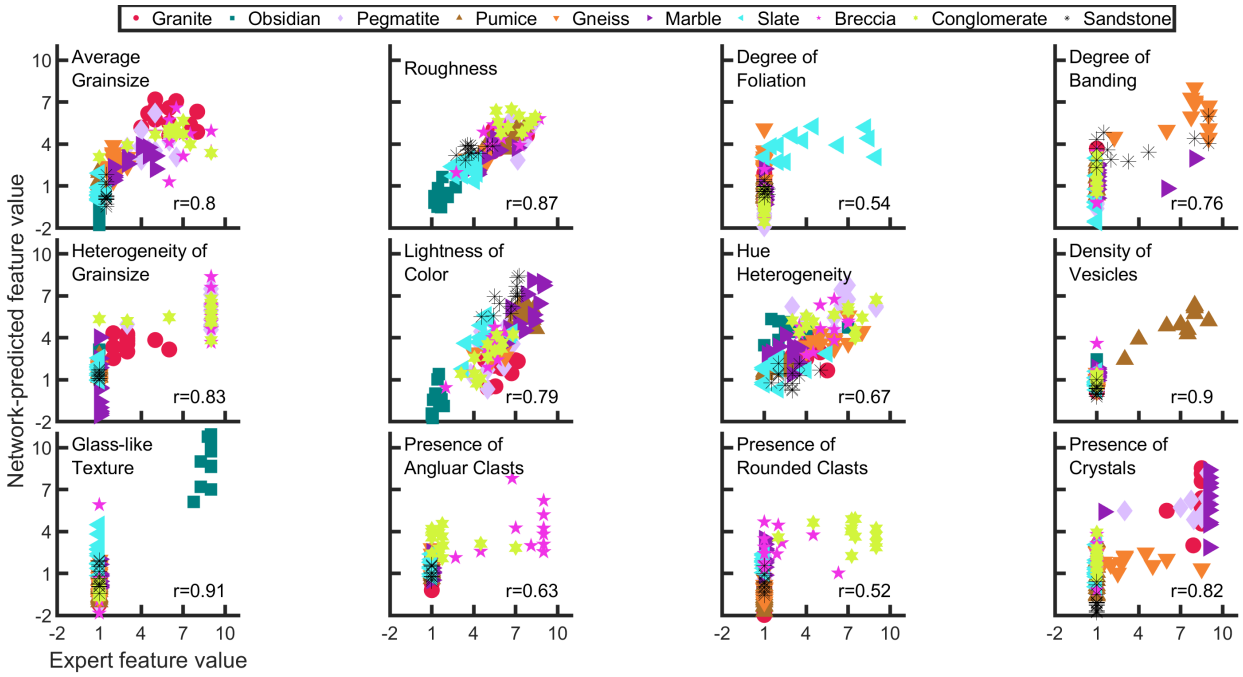


Figure 7.23: Feature Correlation - Hybrid Sequential CBM Vs Expert Ratings - 12 Features - Continuous Crystal Rating

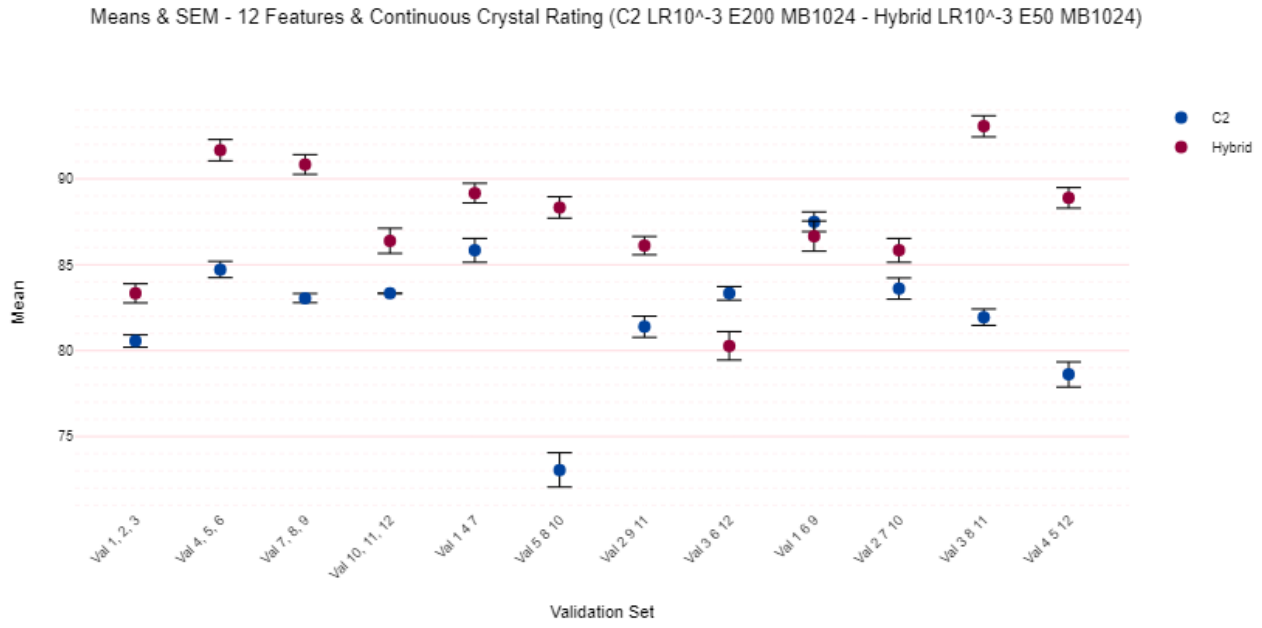


Figure 7.24: Means & SEM - Validation Sets - 12 Features - Continuous Crystal Rating

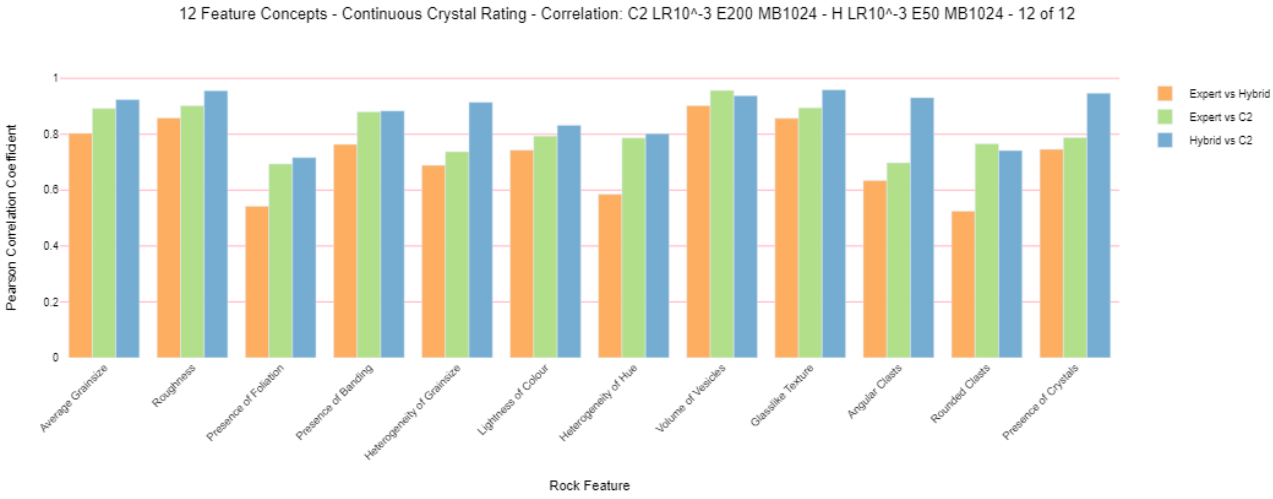


Figure 7.25: 12 Feature Concepts - Continuous Crystal Rating - Mean correlation of Feature Concepts

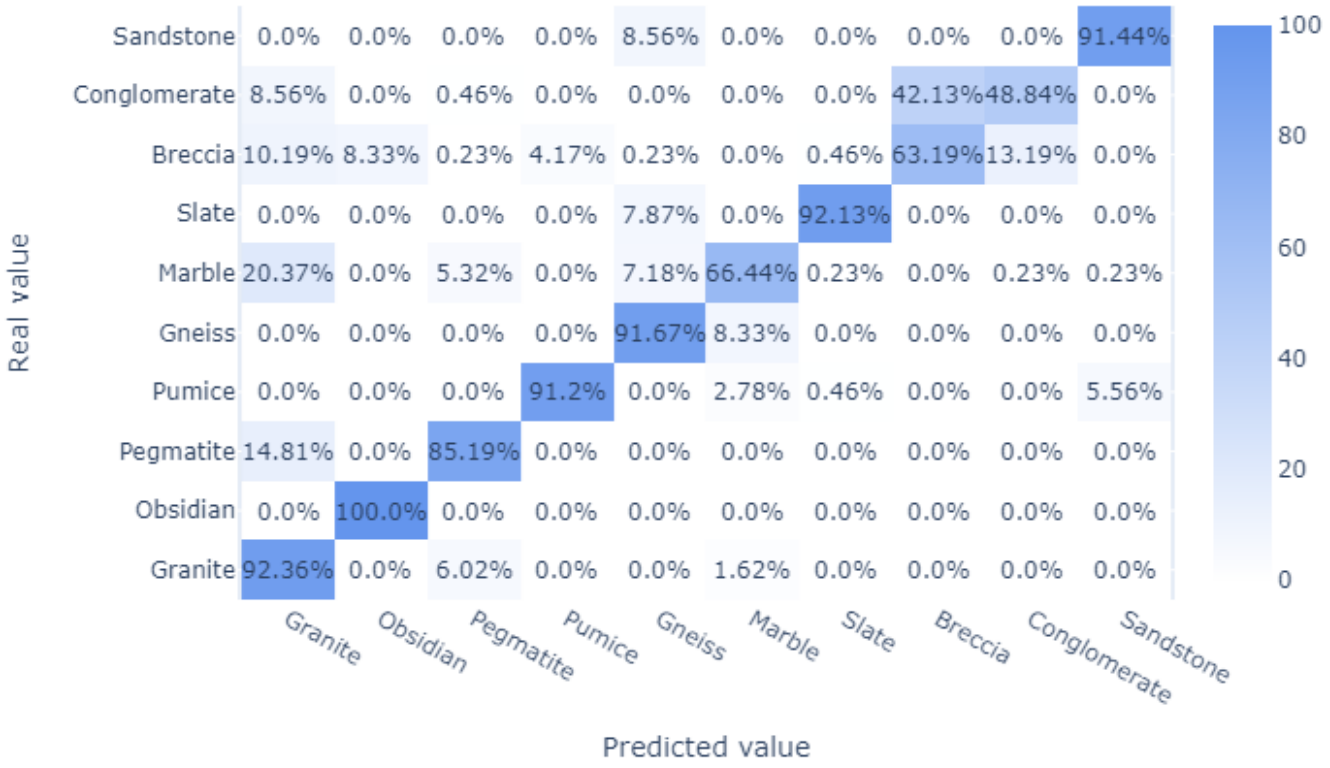


Figure 7.26: Sequential CBM Network Accuracy Confusion Matrix - 12 Features - Continuous Crystal Ratings

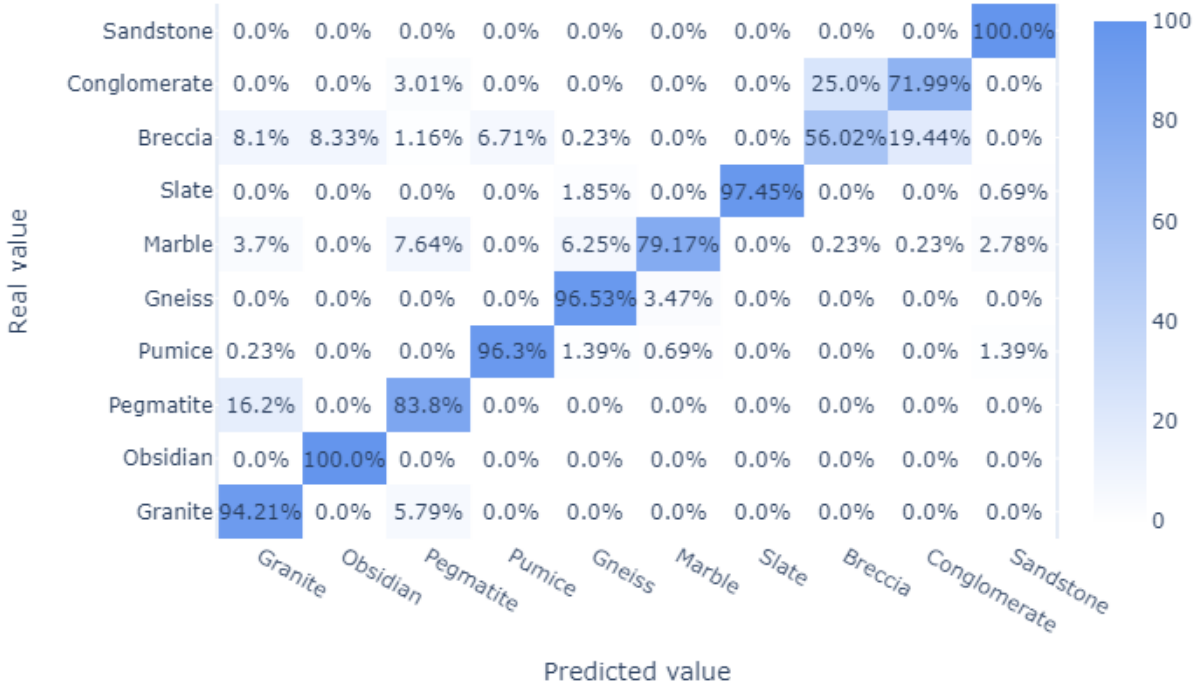


Figure 7.27: Hybrid Network Accuracy Confusion Matrix - 12 Features - Continuous Crystal Ratings

7.3 Tables

7.3.1 An Example of 13 Expert Feature Ratings for Granite Used for Training

Rock Type: Granite

	Rock Image Instance											
Feature	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Average Grainsize	6.5	4	6	4.5	6	5	5	5	7.5	6	8	8
Roughness	6.25	5.3	5.45	5.35	6.9	5.15	5.75	5.05	7.8	6	5.4	5.65
Presence of Foliation	1	1	1	1	1	1	1	1	1	1	1	1
Presence of Banding	1	1	1	1	1	1	1	1	1	1	1	1
Heterogeneity of Grainsize	3	3	6	2	3	3	2	3	5	2	3	3
Lightness of Color	5	7.15	6.25	6	5	5.55	6.45	6.7	4.8	5.5	7	6.5
Heterogeneity of Hue	5	3	4	3.5	4	3.5	3	3	4	5.5	4	4
Heterogeneity of Brightness	4	5	3	5	1.5	2	4	3	3	2	4	2.5
Volume of Vesicles	1	1	1	1	1	1	1	1	1	1	1	1
Glasslike texture	1	1	1	1	1	1	1	1	1	1	1	1
Angular Clasts	1	1	1	1	1	1	1	1	1	1	1	1
Rounded Clasts	1	1	1	1	1	1	1	1	1	1	1	1
Presence of Crystals	8.5	8.5	6	8.5	7.9	8.5	8.5	8.5	8.5	8.5	8.5	8.5

Table 7.1: An Example of 13 Expert Feature Ratings for Granite - As Used for Training

7.3.2 Feature Correlations - 13 Features - Continuous (Scalar) Crystal Rating 12 Runs of 12 Alternating Validation Sets

Set/Runs	**12/12**	**12/12**	**12/12**	**12/12**	**12/12**	**12/12**	**12/12**	**12/12**	**12/12**
C2									
Epochs	**150**	**200**	**200**	**400**	**400**	**400**	**400**	**400**	**400**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**256**	**256**	**256**	**256**	**256**
Correlation - Expert Feature vs C2									
Average Grainsize	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Roughness	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
Presence of Foliation	0.67	0.67	0.67	0.67	0.66	0.67	0.67	0.67	0.67
Presence of Banding	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
Heterogeneity of Grainsize	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Lightness of Colour	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Heterogeneity of Hue	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84	0.84
Heterogeneity of Brightness	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43

Table 7.2 continued from previous page

Volume of Vesicles	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
Glasslike Texture	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Angular Clasts	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
Rounded Clasts	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
Presence of Crystals	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
Mean Correlation Ex Vs C2	**0.8228**	**0.8230**	**0.8229**	**0.8222**	**0.8224**	**0.8232**	**0.8230**	**0.8222**	**0.8230**
Hybrid									
Epochs	**50**	**200**	**50**	**50**	**15**	**25**	**50**	**100**	**200**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**
Correlation - Expert Feature vs Hybrid									
Average Grainsize	0.77	0.71	0.79	0.81	0.84	0.83	0.81	0.77	0.67
Roughness	0.87	0.76	0.86	0.86	0.89	0.88	0.86	0.81	0.71
Presence of Foliation	0.52	0.59	0.55	0.59	0.60	0.61	0.60	0.60	0.65
Presence of Banding	0.74	0.73	0.76	0.80	0.83	0.83	0.81	0.79	0.72
Heterogeneity of Grainsize	0.85	0.82	0.85	0.86	0.88	0.87	0.86	0.84	0.82
Lightness of Colour	0.75	0.49	0.76	0.78	0.85	0.83	0.80	0.70	0.57
Heterogeneity of Hue	0.71	0.67	0.69	0.67	0.74	0.72	0.67	0.61	0.57
Heterogeneity of Brightness	0.27	0.22	0.30	0.32	0.36	0.34	0.32	0.28	0.21
Volume of Vesicles	0.70	0.81	0.76	0.90	0.93	0.93	0.90	0.88	0.85
Glasslike Texture	0.92	0.93	0.93	0.94	0.95	0.95	0.94	0.94	0.93
Angular Clasts	0.58	0.53	0.60	0.62	0.64	0.63	0.61	0.57	0.53
Rounded Clasts	0.52	0.44	0.55	0.61	0.65	0.65	0.64	0.61	0.56
Presence of Crystals	0.83	0.71	0.84	0.85	0.86	0.86	0.85	0.82	0.77
Mean Correlation Ex Vs Hybrid	**0.6944**	**0.6468**	**0.7116**	**0.7385**	**0.7711**	**0.7631**	**0.7442**	**0.7084**	**0.6584**
Correlation - Hybrid vs C2									
Average Grainsize	0.89	0.83	0.91	0.93	0.96	0.95	0.93	0.89	0.80
Roughness	0.94	0.83	0.94	0.95	0.98	0.97	0.95	0.90	0.81
Presence of Foliation	0.72	0.73	0.76	0.84	0.88	0.87	0.85	0.82	0.85
Presence of Banding	0.85	0.83	0.89	0.94	0.97	0.96	0.95	0.91	0.82
Heterogeneity of Grainsize	0.94	0.89	0.94	0.95	0.98	0.97	0.95	0.93	0.90
Lightness of Colour	0.79	0.52	0.81	0.84	0.92	0.91	0.87	0.76	0.61
Heterogeneity of Hue	0.83	0.76	0.81	0.80	0.89	0.86	0.80	0.72	0.67
Heterogeneity of Brightness	0.60	0.42	0.68	0.77	0.85	0.80	0.73	0.61	0.39
Volume of Vesicles	0.71	0.83	0.77	0.92	0.96	0.95	0.92	0.89	0.87
Glasslike Texture	0.97	0.97	0.98	0.98	0.99	0.99	0.99	0.98	0.98
Angular Clasts	0.82	0.82	0.87	0.91	0.94	0.93	0.91	0.88	0.84
Rounded Clasts	0.73	0.61	0.78	0.85	0.92	0.91	0.89	0.84	0.79
Presence of Crystals	0.94	0.76	0.94	0.95	0.98	0.97	0.95	0.90	0.83
Mean Correlation Hybrid Vs C2	**0.8264**	**0.7534**	**0.8535**	**0.8947**	**0.9388**	**0.9256**	**0.8985**	**0.8501**	**0.7801**

Table 7.2: Feature Correlations - 13 Features - Continuous (Scalar) Crystal Rating 12 Runs of 12 Alternating Validation Sets

7.3.3 Feature Correlations - 13 Features - Continuous (Scalar) Crystal Rating 1 Run of 12 Alternating Validation Sets

Set/Runs	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12
C2									
Epochs	**150**	**200**	**200**	**200**	**200**	**200**	**200**	**200**	**100**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.01**
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**
Correlation - Expert Feature vs C2									
Average Grainsize	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Roughness	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
Presence of Foliation	0.66	0.67	0.68	0.66	0.67	0.67	0.67	0.67	0.67
Presence of Banding	0.87	0.88	0.87	0.88	0.87	0.87	0.87	0.88	0.88
Heterogeneity of Grainsize	0.89	0.89	0.89	0.89	0.88	0.88	0.89	0.89	0.89
Lightness of Colour	0.93	0.93	0.92	0.93	0.93	0.93	0.93	0.93	0.93
Heterogeneity of Hue	0.83	0.83	0.83	0.84	0.83	0.83	0.84	0.83	0.83
Heterogeneity of Brightness	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Volume of Vesicles	0.96	0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.96
Glasslike Texture	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Angular Clasts	0.70	0.70	0.70	0.70	0.69	0.69	0.70	0.70	0.70
Rounded Clasts	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
Presence of Crystals	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
Mean Correlation Ex Vs C2	**0.8220**	**0.8228**	**0.8223**	**0.8213**	**0.8203**	**0.8203**	**0.8221**	**0.8231**	**0.8228**
Hybrid									
Epochs	**150**	**200**	**300**	**400**	**200**	**200**	**200**	**200**	**200**
Learning Rate	**0.001**	**0.0009**	**0.0007**						**0.001**
Learning Rate	**10^-3**			**10^-4**	**6^-3**	**7^-3**	**8^-3**	**9^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**
Correlation - Expert Feature vs Hybrid									
Average Grainsize	0.72	0.71	0.70	0.79	0.49	0.57	0.64	0.68	0.69
Roughness	0.81	0.78	0.78	0.85	0.66	0.70	0.64	0.66	0.73
Presence of Foliation	0.52	0.56	0.56	0.53	0.47	0.49	0.56	0.58	0.62
Presence of Banding	0.69	0.66	0.66	0.76	0.55	0.64	0.68	0.69	0.70
Heterogeneity of Grainsize	0.81	0.80	0.80	0.85	0.74	0.76	0.81	0.81	0.82
Lightness of Colour	0.55	0.49	0.52	0.77	0.46	0.45	0.35	0.44	0.52
Heterogeneity of Hue	0.68	0.65	0.63	0.65	0.65	0.66	0.66	0.66	0.56
Heterogeneity of Brightness	0.20	0.18	0.20	0.28	0.07	0.10	0.20	0.20	0.19
Volume of Vesicles	0.64	0.71	0.68	0.75	0.72	0.71	0.80	0.79	0.77
Glasslike Texture	0.92	0.92	0.92	0.92	0.93	0.93	0.92	0.92	0.93
Angular Clasts	0.47	0.47	0.48	0.60	0.50	0.50	0.53	0.50	0.49
Rounded Clasts	0.41	0.45	0.43	0.52	0.44	0.41	0.40	0.43	0.52
Presence of Crystals	0.79	0.78	0.75	0.82	0.75	0.75	0.71	0.70	0.73
Mean Correlation Ex Vs Hybrid	**0.6303**	**0.6274**	**0.6230**	**0.6990**	**0.5725**	**0.5907**	**0.6082**	**0.6211**	**0.6360**
Correlation - Hybrid vs C2									

Table 7.3.1 continued from previous page

Average Grainsize	0.83	0.83	0.82	0.91	0.59	0.68	0.75	0.80	0.81
Roughness	0.88	0.85	0.85	0.93	0.70	0.75	0.69	0.72	0.81
Presence of Foliation	0.69	0.73	0.71	0.75	0.65	0.65	0.71	0.74	0.83
Presence of Banding	0.77	0.74	0.75	0.88	0.65	0.73	0.79	0.79	0.80
Heterogeneity of Grainsize	0.88	0.87	0.88	0.94	0.80	0.82	0.88	0.88	0.89
Lightness of Colour	0.57	0.51	0.55	0.81	0.48	0.47	0.36	0.47	0.56
Heterogeneity of Hue	0.78	0.76	0.72	0.77	0.77	0.79	0.75	0.76	0.67
Heterogeneity of Brightness	0.36	0.37	0.36	0.70	0.11	0.18	0.34	0.36	0.40
Volume of Vesicles	0.64	0.71	0.68	0.75	0.74	0.72	0.82	0.80	0.78
Glasslike Texture	0.97	0.97	0.97	0.97	0.99	0.99	0.97	0.97	0.97
Angular Clasts	0.70	0.73	0.70	0.85	0.74	0.76	0.79	0.80	0.78
Rounded Clasts	0.58	0.63	0.59	0.74	0.59	0.56	0.54	0.59	0.73
Presence of Crystals	0.88	0.85	0.81	0.93	0.82	0.82	0.76	0.74	0.76
Mean Correlation Hybrid Vs C2	**0.7333**	**0.7337**	**0.7226**	**0.8413**	**0.6627**	**0.6862**	**0.7050**	**0.7236**	**0.7529**

Table 7.3.1: Feature Correlations - 13 Features - Continuous (Scalar) Crystal Rating 1 Run of 12 Alternating Validation Sets - Part 1

Set/Runs	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12
C2								
Epochs	**200**	**200**	**200**	**200**	**200**	**400**	**400**	**2000**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**		
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**11^-3**	**10^-4**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**128**	**1024**	**1024**
Correlation - Expert Feature vs C2								
Average Grainsize	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
Roughness	0.92	0.92	0.92	0.92	0.92	0.91	0.92	0.92
Presence of Foliation	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
Presence of Banding	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
Heterogeneity of Grainsize	0.89	0.89	0.89	0.89	0.89	0.73	0.89	0.89
Lightness of Colour	0.93	0.93	0.93	0.93	0.93	0.81	0.93	0.93
Heterogeneity of Hue	0.83	0.83	0.83	0.83	0.83	0.79	0.83	0.83
Heterogeneity of Brightness	0.43	0.43	0.43	0.43	0.43	0.45	0.43	0.43
Volume of Vesicles	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
Glasslike Texture	0.95	0.95	0.95	0.95	0.95	0.89	0.95	0.95
Angular Clasts	0.71	0.69	0.70	0.70	0.70	0.70	0.70	0.70
Rounded Clasts	0.76	0.77	0.76	0.76	0.76	0.76	0.76	0.76
Presence of Crystals	0.88	0.88	0.88	0.88	0.88	0.80	0.88	0.88
Mean Correlation Ex Vs C2	**0.8230**	**0.8226**	**0.8228**	**0.8228**	**0.8231**	**0.7879**	**0.8228**	**0.8228**
Hybrid								
Epochs	**100**	**1000**	**200**	**200**	**200**	**200**	**200**	**200**
Learning Rate			**0.0008**			**0.001**	**0.001**	**0.001**
Learning Rate	**11^-3**	**11^-3**		**6^-4**	**12^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**
Correlation - Expert Feature vs Hybrid								

Table 7.3.2 continued from previous page

Average Grainsize	0.77	0.61	0.72	0.72	0.73	0.66	0.69	0.69
Roughness	0.84	0.73	0.78	0.79	0.80	0.70	0.71	0.68
Presence of Foliation	0.54	0.56	0.56	0.56	0.57	0.63	0.64	0.64
Presence of Banding	0.74	0.63	0.67	0.67	0.74	0.72	0.70	0.70
Heterogeneity of Grainsize	0.84	0.79	0.80	0.81	0.83	0.69	0.80	0.80
Lightness of Colour	0.70	0.48	0.51	0.52	0.67	0.61	0.49	0.50
Heterogeneity of Hue	0.63	0.64	0.64	0.64	0.63	0.48	0.58	0.57
Heterogeneity of Brightness	0.25	0.11	0.19	0.19	0.27	0.18	0.18	0.17
Volume of Vesicles	0.72	0.71	0.70	0.70	0.80	0.85	0.83	0.87
Glasslike Texture	0.93	0.92	0.92	0.92	0.92	0.85	0.92	0.91
Angular Clasts	0.56	0.48	0.48	0.48	0.54	0.50	0.50	0.52
Rounded Clasts	0.52	0.46	0.46	0.46	0.47	0.59	0.52	0.53
Presence of Crystals	0.81	0.76	0.79	0.79	0.73	0.69	0.76	0.77
Mean Correlation Ex Vs Hybrid	**0.6804**	**0.6055**	**0.6324**	**0.6341**	**0.6697**	**0.6263**	**0.6399**	**0.6288**
Correlation - Hybrid vs C2								
Average Grainsize	0.89	0.72	0.84	0.84	0.86	0.78	0.81	0.80
Roughness	0.91	0.78	0.86	0.86	0.87	0.80	0.79	0.77
Presence of Foliation	0.75	0.74	0.74	0.74	0.74	0.84	0.83	0.83
Presence of Banding	0.85	0.70	0.75	0.76	0.86	0.82	0.80	0.80
Heterogeneity of Grainsize	0.93	0.86	0.87	0.88	0.91	0.91	0.87	0.87
Lightness of Colour	0.75	0.50	0.53	0.54	0.71	0.64	0.52	0.52
Heterogeneity of Hue	0.75	0.76	0.76	0.76	0.74	0.67	0.69	0.68
Heterogeneity of Brightness	0.58	0.18	0.39	0.40	0.54	0.37	0.38	0.36
Volume of Vesicles	0.73	0.71	0.70	0.70	0.82	0.85	0.84	0.88
Glasslike Texture	0.98	0.97	0.98	0.98	0.97	0.96	0.97	0.97
Angular Clasts	0.81	0.76	0.73	0.73	0.84	0.81	0.80	0.81
Rounded Clasts	0.73	0.61	0.63	0.64	0.65	0.81	0.71	0.72
Presence of Crystals	0.91	0.82	0.85	0.86	0.78	0.84	0.81	0.82
Mean Correlation Hybrid Vs C2	**0.8119**	**0.7010**	**0.7414**	**0.7439**	**0.7921**	**0.7770**	**0.7556**	**0.7565**

Table 7.3.2: Feature Correlations - 13 Features - Continuous Scalar Crystal Rating 1 Run of 12 Alternating Validation Sets - Part 2

7.3.4 Feature Correlations - 13 Features - Binary Crystal Rating 12 Runs of 12 Alternating Validation Sets

Set/Runs	**12/12**	**12/12**
C2		
Epochs	**200**	**200**
Learning Rate	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**
Correlation - Expert Feature vs C2		

Table 7.4 continued from previous page

Average Grainsize	0.89	0.89
Roughness	0.91	0.91
Presence of Foliation	0.66	0.67
Presence of Banding	0.88	0.88
Heterogeneity of Grainsize	0.74	0.74
Lightness of Colour	0.81	0.80
Heterogeneity of Hue	0.79	0.79
Heterogeneity of Brightness	0.46	0.46
Volume of Vesicles	0.96	0.96
Glasslike Texture	0.89	0.89
Angular Clasts	0.71	0.71
Rounded Clasts	0.76	0.76
Presence of Crystals	0.88	0.88
Mean Correlation Ex Vs C2	**0.7951**	**0.7951**
Hybrid		
Epochs	**15**	**50**
Learning Rate	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**
Correlation - Expert Feature vs Hybrid		
Average Grainsize	0.83	0.81
Roughness	0.89	0.86
Presence of Foliation	0.58	0.56
Presence of Banding	0.80	0.77
Heterogeneity of Grainsize	0.72	0.71
Lightness of Colour	0.79	0.74
Heterogeneity of Hue	0.64	0.57
Heterogeneity of Brightness	0.34	0.26
Volume of Vesicles	0.85	0.79
Glasslike Texture	0.88	0.88
Angular Clasts	0.66	0.63
Rounded Clasts	0.57	0.48
Presence of Crystals	0.85	0.84
Mean Correlation Ex Vs Hybrid	**0.7231**	**0.6837**
Correlation - Hybrid vs C2		
Average Grainsize	0.95	0.92
Roughness	0.97	0.93
Presence of Foliation	0.85	0.79
Presence of Banding	0.94	0.90
Heterogeneity of Grainsize	0.97	0.95
Lightness of Colour	0.93	0.83
Heterogeneity of Hue	0.84	0.76
Heterogeneity of Brightness	0.79	0.60
Volume of Vesicles	0.87	0.80
Glasslike Texture	0.99	0.99

Table 7.4 continued from previous page

Angular Clasts	0.94	0.91
Rounded Clasts	0.82	0.69
Presence of Crystals	0.98	0.97
Mean Correlation Hybrid Vs C2	**0.9112**	**0.8496**

Table 7.4: Feature Correlations - 13 Features - Binary Crystal Rating 12 Runs of 12 Alternating Validation Sets

7.3.5 13 Features - Accuracy - Binary vs Crystal Rating 12 Runs of 12 Alternating Validation Sets + Run with Re-Rated Data

	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous	Continuous	Binary Crystal	Binary Crystal
Set/Runs	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12
C2									Re-Rated data
Epochs	**150**	**200**	**400**	**400**	**400**	**400**	**400**	**200**	**200**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**256**	**256**	**256**	**256**	**256**	**1024**	**1024**
Accuracy Mean - Val Set	**84.05%**	**83.66%**	83.26%	83.06%	83.06%	83.36%	83.47%	**84.63%**	**83.37%**
SEM - Val Set	**0.49%**	**0.52%**	0.66%	0.67%	0.65%	0.61%	0.67%	**0.53%**	**0.49%**
Accuracy Mean - Mean Set	**84.05%**	**83.66%**						**84.63%**	**83.33%**
STD of Means (Mean Set)	**0.77%**	**0.57%**						**4.03%**	**2.75%**
SEM (Mean Set)	**0.77%**	**0.16%**						**1.16%**	**0.79%**
Hybrid									
Epochs	**50**	**50**	**15**	**25**	**50**	**100**	**200**	**50**	**50**
Learning Rate	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**	**1024**
Accuracy Mean - From Val Set	**87.25%**	**87.80%**	86.50%	87.06%	87.31%	87.55%	87.73%	**87.59%**	**86.40%**
SEM - Val Set	**0.71%**	**0.68%**	0.73%	0.72%	0.67%	0.80%	0.76%	**0.70%**	**0.73%**
Accuracy Mean - Mean Set	**87.25%**	**87.80%**			87.22%			**87.59%**	**86.41%**
STD of Means (Mean Set)	**0.78%**	**0.58%**			0.78%			**2.94%**	**2.69%**
SEM (Mean Set)	**0.78%**	**0.17%**			0.78%			**0.85%**	**0.78%**

Table 7.5: 13 Features - Accuracy - Binary vs Crystal Rating 12 Runs of 12 Alternating Validation Sets + Run with Re-Rated Data

7.3.6 12 Features - Accuracy - Binary vs Crystal Rating 12 Runs of 12 Alternating Validation Sets

	Binary Crystal	Continuous
Set/Runs	12/12	12/12
C2		
Epochs	**200**	**200**
Learning Rate	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**
Accuracy Mean - Val Set	**82.48%**	**82.25%**
SEM - Val Set	**0.57%**	**0.52%**
Accuracy Mean - Run Set	**82.48%**	**82.25%**
STD of Means (Run Set)	**2.79%**	**3.57%**
SEM (RunSet)	**0.80%**	**1.03%**
Hybrid		
Epochs	**50**	**50**
Learning Rate	**0.001**	**0.001**
Learning Rate	**10^-3**	**10^-3**
Minibatch Size	**1024**	**1024**

Table 7.6 continued from previous page

Accuracy Mean - From Val Set	**88.36%**	**87.55%**
SEM - Val Set	**0.62%**	**0.65%**
Accuracy Mean - Run Set	**88.36%**	**87.55%**
STD of Means (Run Set)	**3.41%**	**3.43%**
SEM (RunSet)	**0.98%**	**0.99%**

Table 7.6: 12 Features - Accuracy - Binary vs Crystal Rating 12 Runs of 12 Alternating Validation Sets

7.3.7 Weights and calculated L2 regularisation values from a single run of a validation set (Images 1,2,3). Sequential CBM, 13 Features and binary rated crystals

	Average Grainsize	Roughness	Presence of Foliation	Presence of Banding	Heterogeneity of Grainsize	Lightness of Colour	Heterogeneity of Hue	Heterogeneity of Brightness	Volume of Vesicles	Glasslike texture	Angular Clasts	Rounded Clasts	Presence of Crystals	Total Sum L2
Granite	1.5372	-0.0603	0.4126	-0.8737	0.0209	-0.6585	-0.8273	-0.6254	-0.3221	-0.1575	-0.5754	-0.1742	0.8458	**6.0152**
L2	**2.3629**	0.0036	**0.1703**	**0.7634**	0.0004	0.4336	**0.6844**	0.3911	0.1038	0.0248	0.3311	0.0303	**0.7154**	
Obsidian	0.0217	-0.6697	0.0375	-0.1029	-0.2922	-1.0776	0.0674	0.2991	0.0825	0.8646	0.0871	0.0538	0.0194	**2.5668**
L2	0.0005	0.4486	0.0014	0.0106	0.0854	1.1612	0.0045	0.0895	0.0068	**0.7475**	0.0076	0.0029	0.0004	
Pegmatite	-0.5469	0.7885	-1.5285	-0.6705	1.3582	-0.1488	1.3095	-1.1142	-0.6384	-0.4296	-1.5046	-1.6249	0.1343	**14.0439**
L2	0.2991	**0.6218**	**2.3362**	0.4496	**1.8447**	0.0221	**1.7147**	**1.2415**	0.4076	0.1846	**2.2639**	**2.6401**	0.0180	
Pumice	-1.7590	0.7298	-0.6836	-0.1830	-0.0670	1.1350	-0.8926	-1.8654	0.9646	-0.7853	-0.8974	-0.7868	-0.3236	**12.7726**
L2	**3.0941**	0.5326	0.4673	0.0335	0.0045	**1.2882**	**0.7967**	**3.4797**	**0.9304**	0.6166	0.8053	0.6190	0.1047	
Gneiss	-1.0560	-0.0175	1.0860	1.0367	-1.6573	-1.1296	0.1464	1.1764	0.2793	-0.9837	-0.6134	-1.0950	0.8857	**12.2029**
L2	**1.1152**	0.0003	**1.1795**	**1.0748**	**2.7465**	**1.2759**	0.0214	**1.3839**	0.0780	**0.9677**	0.3762	**1.1989**	**0.7845**	
Marble	0.3165	-0.3164	-1.5363	0.0066	-3.4487	0.2820	0.1255	-0.2597	-1.0303	-0.1758	-0.1091	1.6845	1.0421	**19.6444**
L2	0.1002	0.1001	**2.3602**	0.0000	**11.8932**	0.0795	0.0158	0.0674	**1.0616**	0.0309	0.0119	**2.8376**	**1.0860**	
Slate	-1.7826	-1.3018	2.4846	-0.6862	0.3148	0.3701	-1.8097	0.8151	-0.4503	-0.5997	0.1608	1.3792	0.0669	**18.1869**
L2	**3.1775**	**1.6948**	**6.1733**	0.4708	0.0991	0.1370	**3.2751**	**0.6643**	0.2027	0.3596	0.0259	**1.9022**	0.0045	
Breccia	0.8250	-0.5047	-1.1521	-0.9379	1.4214	-0.6848	0.2302	0.2953	-0.3709	-0.7318	0.8035	-0.8183	-1.1566	**9.0980**
L2	**0.6806**	0.2547	**1.3274**	**0.8797**	**2.0204**	0.4689	0.0530	0.0872	0.1376	0.5355	**0.6455**	**0.6696**	**1.3378**	
Conglomerate	0.3192	1.3012	-0.5298	-0.0828	-0.6819	-0.6468	0.3138	-0.7170	-0.9182	-0.8454	0.0509	1.5011	-0.5944	**7.7457**
L2	0.1019	**1.6931**	0.2806	0.0069	0.4650	0.4184	0.0985	0.5141	0.8431	0.7148	0.0026	**2.2533**	0.3534	
Sandstone	-0.0951	0.4521	-1.2946	0.3708	-0.5835	1.3722	-0.3096	0.1394	-1.9961	-0.6579	-1.1462	-0.3137	-0.9120	**11.0264**
L2	0.0090	0.2044	**1.6759**	0.1375	0.3404	**1.8828**	0.0958	0.0194	**3.9843**	0.4329	**1.3137**	0.0984	**0.8318**	

Key							
L2 Value	**>= 0.5**	**>= 1**	**>= 1.5**	**>= 2**	**>= 2.5**	**>= 3**	**>= 3.5**

Table 7.7: Weights and calculated L2 regularisation values from a single run of a validation set (Images 1,2,3). Sequential CBM, 13 Features and binary rated crystals

7.3.8 Analysis of L2 regularisation values for assessment of the complexity of concepts. Data from a single run of a validation set (Images 1,2,3). Sequential CBM, 13 Features and binary rated crystals

	Total Sum L2	Sum L2 >= 0.5	L2 >=0.5	% L2 >= 0.5	Sum L2 >= 1	L2 >= 1	% L2 >= 1	Sum L2 >= 1.5	L2 >= 1.5	% L2 >= 1.5	Sum L2 >= 2	L2 >= 2	% L2 >= 2	Sum L2 >= 2.5	L2 >= 2.5	% L2 >= 2.5	Sum L2 >= 3	L2 >= 3	% L2 >= 3	Sum L2 >= 3.5	L2 >= 3.5	% L2 >= 3.5
Granite	**6.0152**																					
		4.5260	4	75.24%	2.3629	1	39.28%	2.3629	1	39.28%	**2.3629**	**1**	**39.28%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Obsidian	**2.5668**																					
		1.9087	2	74.36%	**1.1612**	**1**	**45.24%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Pegmatite	**14.0439**																					
		12.6629	7	90.17%	12.0411	6	85.74%	10.7997	5	76.90%	7.2403	3	51.55%	**2.6401**	**1**	**18.80%**	0.0000	0	0.00%	0.0000	0	0.00%
Pumice	**12.7726**																					
		12.1625	9	95.22%	7.8620	3	61.55%	6.5738	2	51.47%	6.5738	2	51.47%	6.5738	2	51.47%	**6.5738**	**2**	**51.47%**	0.0000	0	0.00%
Gneiss	**12.2029**																					
		11.7269	9	96.10%	9.9748	7	81.74%	2.7465	1	22.51%	2.7465	1	22.51%	**2.7465**	**1**	**22.51%**	0.0000	0	0.00%	0.0000	0	0.00%
Marble	**19.6444**																					
		19.2386	5	97.93%	19.2386	5	97.93%	17.0910	3	87.00%	17.0910	3	87.00%	14.7308	2	74.99%	11.8932	1	60.54%	**11.8932**	**1**	**60.54%**
Slate	**18.1869**																					
		16.8873	6	92.85%	16.2230	5	89.20%	16.2230	5	89.20%	12.6259	3	69.42%	12.6259	3	69.42%	12.6259	3	69.42%	**6.1733**	**1**	**33.94%**
Breccia	**9.0980**																					
		8.0966	8	88.99%	4.6856	3	51.50%	2.0204	1	22.21%	**2.0204**	**1**	**22.21%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Conglomerate	**7.7457**																					
		6.0185	5	77.70%	3.9465	2	50.95%	3.9465	2	50.95%	**2.2533**	**1**	**29.09%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Sandstone	**11.0264**																					
		9.6885	5	87.87%	8.8567	4	80.32%	7.5430	3	68.41%	3.9843	1	36.13%	3.9843	1	36.13%	3.9843	1	36.13%	**3.9843**	**1**	**36.13%**

Table 7.8: Analysis of L2 regularisation values for assessment of the complexity of concepts. Data from a single run of a validation set (Images 1,2,3). Sequential CBM, 13 Features and binary rated crystals

7.3.9 Weights and calculated L2 regularisation values from a single run of a validation set (Images 1,2,3). Hybrid Classifier CBM, 13 Features and binary rated crystals

	Average Grainsize	Roughness	Presence of Foliation	Presence of Banding	Heterogeneity of Grainsize	Lightness of Colour	Heterogeneity of Hue	Heterogeneity of Brightness	Volume of Vesicles	Glasslike texture	Angular Clasts	Rounded Clasts	Presence of Crystals	Total Sum L2
Granite	1.5398	-0.0618	0.3249	-0.8968	0.0142	-0.6557	-0.8349	-0.6013	-0.2076	-0.1827	-0.5693	-0.2712	0.8495	**5.9691**
L2	**2.3709**	0.0038	**0.1055**	**0.8043**	0.0002	0.4299	**0.6970**	0.3616	0.0431	0.0334	0.3241	0.0735	**0.7217**	
Obsidian	-0.1441	-0.4293	-0.0051	-0.0770	-0.1201	-0.6695	0.0621	0.2476	0.0584	0.8779	-0.0135	-0.0224	-0.1382	**1.5327**
L2	0.0208	0.1843	0.0000	0.0059	0.0144	0.4483	0.0039	0.0613	0.0034	**0.7707**	0.0002	0.0005	0.0191	
Pegmatite	-0.5469	0.7619	-1.0666	-0.6756	1.2888	-0.1353	1.2747	-1.0046	-0.6336	-0.2767	-1.3208	-1.3582	0.1432	**10.8745**
L2	0.2991	**0.5805**	**1.1375**	0.4564	**1.6609**	0.0183	**1.6248**	**1.0093**	0.4015	0.0766	**1.7444**	**1.8447**	0.0205	
Pumice	-1.2677	0.6754	-0.3592	-0.0694	-0.1608	1.0844	-0.7213	-1.6140	0.9841	-0.4334	-0.5185	-0.4529	-0.2507	**8.2170**
L2	**1.6069**	0.4561	0.1290	0.0048	0.0259	**1.1759**	**0.5202**	**2.6049**	**0.9685**	0.1878	0.2689	0.2051	0.0629	
Gneiss	-0.9959	-0.0135	0.8578	1.0416	-1.3591	-1.0625	0.1636	1.1312	0.2480	-0.8989	-0.5094	-0.8219	0.8508	**9.6233**
L2	**0.9918**	0.0002	**0.7359**	**1.0850**	**1.8471**	**1.1289**	0.0268	**1.2795**	0.0615	**0.8080**	0.2594	**0.6755**	**0.7238**	
Marble	0.3034	-0.3141	-1.2376	0.0111	-3.0005	0.2863	0.1422	-0.2782	-0.9398	-0.0437	-0.0104	1.4457	1.0608	**15.0057**
L2	0.0920	0.0987	**1.5317**	0.0001	**9.0029**	0.0819	0.0202	0.0774	**0.8833**	0.0019	0.0001	**2.0900**	**1.1253**	
Slate	-1.3643	-1.1845	2.0082	-0.6800	0.1519	0.3542	-1.4236	0.7363	-0.4191	-0.4691	0.1223	1.0709	0.0016	**12.0342**
L2	**1.8612**	**1.4031**	**4.0327**	0.4623	0.0231	0.1255	**2.0266**	**0.5421**	0.1757	0.2201	0.0150	**1.1467**	0.0000	
Breccia	0.7365	-0.4426	-0.7717	-0.7348	1.3404	-0.5896	0.2456	0.2723	-0.2675	-0.5917	0.8421	-0.7934	-1.0474	**7.0099**
L2	**0.5425**	0.1959	**0.5956**	**0.5399**	**1.7966**	0.3477	0.0603	0.0741	0.0716	0.3501	**0.7091**	**0.6295**	**1.0970**	
Conglomerate	0.2620	1.1999	-0.1838	-0.0350	-0.6304	-0.6299	0.2373	-0.6637	-0.6672	-0.5611	-0.0206	1.4839	-0.5962	**6.1521**
L2	0.0686	**1.4396**	0.0338	0.0012	0.3974	0.3968	0.0563	0.4405	0.4451	0.3148	0.0004	**2.2019**	0.3555	
Sandstone	-0.0618	0.4383	-0.9773	0.3775	-0.4186	1.3168	-0.2654	0.1865	-1.7667	-0.5471	-0.9480	-0.3055	-1.0153	**8.7511**
L2	0.0038	0.1921	**0.9550**	0.1425	0.1752	**1.7339**	0.0704	0.0348	**3.1212**	0.2993	**0.8987**	0.0934	**1.0308**	

Key							
L2 Value	**>= 0.5**	**>= 1**	**>= 1.5**	**>= 2**	**>= 2.5**	**>= 3**	**>= 3.5**

Table 7.9: Weights and calculated L2 regularisation values from a single run of a validation set (Images 1,2,3). Hybrid Classifier CBM, 13 Features and binary rated crystals

7.3.10 Analysis of L2 regularisation values for assessment of the complexity of concepts. Data from a single run of a validation set (Images 1,2,3). Hybrid Classifier CBM, 13 Features and binary rated crystals

	Total Sum L2	Sum L2 <0.5	L2 <0.5	% L2 <0.5	Sum L2 <1	L2 <1	% L2 <1	Sum L2 <1.5	L2 <1.5	% L2 <1.5	Sum L2 <2	L2 <2	% L2 <2	Sum L2 <2.5	L2 <2.5	% L2 <2.5	Sum L2 <3	L2 <3	% L2 <3	Sum L2 <3.5	L2 <3.5	% L2 <3.5
Granite	**5.9691**																					
		4.5939	4	76.96%	2.3709	1	39.72%	2.3709	1	39.72%	**2.3709**	**1**	**39.72%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Obsidian	**1.5327**																					
		0.7707	**1**	**50.28%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Pegmatite	**10.8745**																					
		9.6021	7	88.30%	9.0216	6	82.96%	**6.8748**	**4**	**63.22%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Pumice	**8.2170**																					
		6.8765	5	83.69%	5.3877	3	65.57%	4.2119	2	51.26%	2.6049	1	31.70%	**2.6049**	**1**	**31.70%**	0.0000	0	0.00%	0.0000	0	0.00%
Gneiss	**9.6233**																					
		9.2754	9	96.38%	5.3406	4	55.50%	1.8471	1	19.19%	0.0000	0	0.00%	**0.0000**	**0**	**0.00%**	0.0000	0	0.00%	0.0000	0	0.00%
Marble	**15.0057**																					
		14.6333	5	97.52%	13.7500	4	91.63%	12.6247	3	84.13%	11.0929	2	73.92%	9.0029	1	60.00%	9.0029	1	60.00%	**9.0029**	**1**	**60.00%**
Slate	**12.0342**																					
		11.0126	6	91.51%	10.4705	5	87.01%	7.9206	3	65.82%	6.0593	2	50.35%	4.0327	1	33.51%	4.0327	1	33.51%	**4.0327**	**1**	**33.51%**
Breccia	**7.0099**																					
		5.9102	7	84.31%	2.8936	2	41.28%	**1.7966**	**1**	**25.63%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Conglomerate	**6.1521**																					
		3.6416	2	59.19%	3.6416	2	59.19%	2.2019	1	35.79%	**2.2019**	**1**	**35.79%**	0.0000	0	0.00%	0.0000	0	0.00%	0.0000	0	0.00%
Sandstone	**8.7511**																					
		7.7396	5	88.44%	5.8858	3	67.26%	4.8551	2	55.48%	3.1212	1	35.67%	3.1212	1	35.67%	**3.1212**	**1**	**35.67%**	0.0000	0	0.00%

Table 7.10: Analysis of L2 regularisation values for assessment of the complexity of concepts. Data from a single run of a validation set (Images 1,2,3). Hybrid Classifier CBM, 13 Features and binary rated crystals

7.3.11 Analysing the complexity of concepts between the sequential CBM and Hybrid Classifier CBM using summed L2 values for all concepts. Data from a single run of a validation set (Images 1,2,3)

	sCBM	Hybrid
All Concepts	**Sum L2**	**Sum L2**
Granite	6.0152	5.9691
Obsidian	2.5668	1.5327
Pegmatite	14.0439	10.8745
Pumice	12.7726	8.2170
Gneiss	12.2029	9.6233
Marble	19.6444	15.0057
Slate	18.1869	12.0342
Breccia	9.0980	7.0099
Conglomerate	7.7457	6.1521
Sandstone	11.0264	8.7511
Total L2	**113.3028**	**85.1696**

Table 7.11: Analysing the complexity of concepts between the sequential CBM and Hybrid Classifier CBM using summed L2 values for all concepts. Data from a single run of a validation set (Images 1,2,3)

References

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity Checks for Saliency Maps," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/294a8ed24b1ad22ec2e7efea049b8737-Abstract.html (cit. on pp. 23, 26).

[2] H. Alkaissi and S. I. McFarlane, "Artificial Hallucinations in ChatGPT: Implications in Scientific Writing," *Cureus*, Feb. 19, 2023. DOI: 10.7759/cureus.35179. [Online]. Available: https://www.cureus.com/articles/138667-artificial-hallucinations-in-chatgpt-implications-in-scientific-writing (cit. on p. 12).

[3] K. Amarasinghe, K. T. Rodolfa, S. Jesus, *et al.* "On the Importance of Application-Grounded Experimental Design for Evaluating Explainable ML Methods." arXiv: 2206.13503 [cs]. (Feb. 21, 2023), [Online]. Available: http://arxiv.org/abs/2206.13503, preprint (cit. on p. 14).

[4] O. D. Apuke and B. Omar, "Fake news and COVID-19: Modelling the predictors of fake news sharing among social media users," *Telematics and Informatics*, vol. 56, p. 101 475, Jan. 2021. DOI: 10.1016/j.tele.2020.101475. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0736585320301349 (cit. on p. 13).

[5] G. Ashby, L. A. Alfonso-Reese, U. Turken, and E. M. Waldron, "A Neuropsychological Theory of Multiple Systems in Category Learning," *Psychological Review*, vol. 105, no. 3, p. 442, 1998. DOI: https://psycnet.apa.org/doi/10.1037/0033-295X.105.3.442 (cit. on p. 19).

[6] S. Attewell. "Exploring the role of generative AI in assessments: A student perspective," National centre for AI. (Jun. 14, 2023), [Online]. Available: https://nationalcentreforai.jiscinvolve.org/wp/2023/06/14/exploring-the-role-of-generative-ai-in-assessments-a-student-perspective/ (cit. on p. 12).

[7] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020. DOI: 10.1016/j.inffus.2019.12.012. [Online].

Available: https://linkinghub.elsevier.com/retrieve/pii/S1566253519308103 (cit. on pp. 14, 15).

[8] C. Benn and S. Lazar, "What's Wrong with Automated Influence," *Canadian Journal of Philosophy*, vol. 52, no. 1, pp. 125–148, Jan. 2022. DOI: 10.1017/can.2021.23. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0045509121000230/type/journal_article (cit. on p. 16).

[9] F. Cabitza, A. Campagner, G. Malgieri, *et al.*, "Quod erat demonstrandum? - Towards a typology of the concept of explanation for the design of explainable AI," *Expert Systems with Applications*, vol. 213, p. 118 888, Mar. 2023. DOI: 10.1016/j.eswa.2022.118888. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417422019066 (cit. on p. 15).

[10] "ChatGPT." (), [Online]. Available: https://chat.openai.com (cit. on p. 11).

[11] Z. Chen, Y. Bei, and C. Rudin, "Concept Whitening for Interpretable Image Recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, Dec. 7, 2020. DOI: 10.1038/s42256-020-00265-z. arXiv: 2002.01650 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2002.01650 (cit. on pp. 22, 24).

[12] D. Coldewey. "Bulgaria now requires (some) government software to be open source," TechCrunch. (Jul. 6, 2016), [Online]. Available: https://techcrunch.com/2016/07/05/bulgaria-now-requires-some-government-software-to-be-open-source/ (cit. on p. 12).

[13] K. M. Collins, M. Barker, M. Espinosa Zarlenga, *et al.*, "Human Uncertainty in Concept-Based AI Systems," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, Montr\'{e}al QC Canada: ACM, Aug. 8, 2023, pp. 869–889. DOI: 10.1145/3600211.3604692. [Online]. Available: https://dl.acm.org/doi/10.1145/3600211.3604692 (cit. on p. 17).

[14] "DALL·E 2." (), [Online]. Available: https://openai.com/dall-e-2 (cit. on p. 11).

[15] R. Daneshjou, M. Yuksekgonul, Z. R. Cai, R. Novoa, and J. Zou, "SkinCon: A skin disease dataset densely annotated by domain experts for fine-grained model debugging and analysis," in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. DOI: 10.48550/arXiv.2302.00785 (cit. on pp. 26, 28, 33).

[16] D. Danks, "Governance via Explainability," in *The Oxford Handbook of AI Governance*, J. B. Bullock, Y.-C. Chen, J. Himmelreich, *et al.*, Eds., 1st ed., Oxford University Press, Feb. 14, 2022. DOI: 10.1093/oxfordhb/9780197579329.013.11. [Online]. Available: https://academic.oup.com/edited-volume/41989/chapter/355437387 (cit. on p. 14).

[17] "ElevenLabs - Generative AI Text to Speech & Voice Cloning." (), [Online]. Available: https://elevenlabs.io/speech-synthesis (cit. on p. 11).

[18] D. C. Elton, "Self-explaining AI as an Alternative to Interpretable AI," in *Artificial General Intelligence*, B. Goertzel, A. I. Panov, A. Potapov, and R. Yampolskiy, Eds., vol. 12177, Cham: Springer International Publishing, 2020, pp. 95–106. DOI: 10.1007/978-3-030-52152-3_10. [Online]. Available: http://link.springer.com/10.1007/978-3-030-52152-3_10 (cit. on p. 22).

[19] E. Erdfelder, F. Faul, and A. Buchner, "GPOWER: A general power analysis program," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 1, pp. 1–11, Mar. 1996. DOI: 10.3758/BF03203630. [Online]. Available: http://link.springer.com/10.3758/BF03203630 (cit. on p. 17).

[20] M. A. Erickson and J. K. Kruschke, "Rules and Exemplars in Category Learning," *Journal of Experimental Psychology General*, vol. 127, 1996. DOI: 10.1037/0096-3445.127.2.107. [Online]. Available: https://www.researchgate.net/publication/13661419_Rules_and_Exemplars_in_Category_Learning (cit. on p. 19).

[21] European Union Intellectual Property Office., *Study on the Impact of Artificial Intelligence on the Infringement and Enforcement of Copyright and Designs*. LU: Publications Office, 2022. [Online]. Available: https://data.europa.eu/doi/10.2814/062663 (cit. on p. 12).

[22] G. Evans. "Climate crisis paintings by famous artists using AI," Ken Bromley Art Supplies. (Dec. 1, 2022), [Online]. Available: https://www.artsupplies.co.uk/blog/climate-crisis-paintings-by-famous-artists-using-ai/ (cit. on p. 11).

[23] F. T. C. C. Advice. "Scammers use AI to enhance their family emergency schemes," Consumer Advice. (Mar. 17, 2023), [Online]. Available: https://consumer.ftc.gov/consumer-alerts/2023/03/scammers-use-ai-enhance-their-family-emergency-schemes (cit. on p. 11).

[24] R. C. Fong and A. Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 3449–3457. DOI: 10.1109/ICCV.2017.371. [Online]. Available: http://ieeexplore.ieee.org/document/8237633/ (cit. on p. 22).

[25] A. Gilson, C. W. Safranek, T. Huang, *et al.*, "How Does ChatGPT Perform on the United States Medical Licensing Examination? The Implications of Large Language Models for Medical Education and Knowledge Assessment," *JMIR Medical Education*, vol. 9, e45312, Feb. 8, 2023. DOI: 10.2196/45312. [Online]. Available: https://mededu.jmir.org/2023/1/e45312 (cit. on p. 29).

[26] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, no. 3, pp. 50–57, Sep. 2017. DOI: 10.1609/aimag.v38i3.2741. arXiv: 1606.08813 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1606.08813 (cit. on pp. 7, 12, 14).

[27] Google. "Klimt vs. Klimt — Google Arts & Culture." (2021), [Online]. Available: https://artsandculture.google.com/project/klimt-vs-klimt (cit. on p. 11).

[28] J. A. Grange, H. Princis, T. R. Kozlowski, *et al.*, "XAI & I: Self-explanatory AI facilitating mutual understanding between AI and human experts," *Procedia Computer Science*, vol. 207, pp. 3600–3607, 2022. DOI: 10.1016/j.procs.2022.09.419. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050922013114 (cit. on pp. 7, 8, 16, 18, 19, 21, 22, 24, 31–35, 45, 46).

[29] K. G. Greene, "AI Governance Multi-stakeholder Convening," in *The Oxford Handbook of AI Governance*, J. B. Bullock, Y.-C. Chen, J. Himmelreich, *et al.*, Eds., 1st ed., Oxford University Press, Mar. 18, 2022. DOI: 10.1093/oxfordhb/9780197579329.013.6. [Online]. Available: https://academic.oup.com/edited-volume/41989/chapter/355436921 (cit. on p. 16).

[30] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–42, Sep. 30, 2019. DOI: 10.1145/3236009. [Online]. Available: https://dl.acm.org/doi/10.1145/3236009 (cit. on p. 15).

[31] D. Gunning, "Explainable Artificial Intelligence (XAI)," 2017. [Online]. Available: https://sites.cc.gatech.edu/~alanwags/DLAI2016/(Gunning)%20IJCAI-16%20DLAI%20WS.pdf (cit. on p. 10).

[32] Z. Guo, M. Schlichtkrull, and A. Vlachos, "A Survey on Automated Fact-Checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, Feb. 9, 2022. DOI: 10.1162/tacl_a_00454. [Online]. Available: https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00454/109469/A-Survey-on-Automated-Fact-Checking (cit. on p. 13).

[33] V. Hassija, V. Chamola, A. Mahapatra, *et al.*, "Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence," *Cognitive Computation*, Aug. 24, 2023. DOI: 10.1007/s12559-023-10179-8. [Online]. Available: https://link.springer.com/10.1007/s12559-023-10179-8 (cit. on p. 14).

[34] M. Havasi, S. Parbhoo, and F. Doshi-Velez, "Addressing Leakage in Concept Bottleneck Models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 386–23 397, Dec. 6, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/944ecf65a46feb578a43abfd5cddd960-Abstract-Conference.html (cit. on pp. 22, 24–27, 31, 33).

[35] R. R. Hoffman, M. Jalaeian, C. Tate, G. Klein, and S. T. Mueller, "Evaluating machine-generated explanations: A "Scorecard" method for XAI measurement science," *Frontiers in Computer Science*, vol. 5, p. 1 114 806, May 9, 2023. DOI: 10.3389/fcomp.2023.1114806. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1114806/full (cit. on pp. 14, 16).

[36] R. R. Hoffman, S. T. Mueller, G. Klein, M. Jalaeian, and C. Tate, "Explainable AI: Roles and stakeholders, desirements and challenges," *Frontiers in Computer Science*, vol. 5, p. 1 117 848, Aug. 17, 2023. DOI: 10.3389/fcomp.2023.1117848. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1117848/full (cit. on p. 14).

[37] R. R. Hoffman, T. Miller, G. Klein, S. T. Mueller, and W. J. Clancey, "Increasing the Value of XAI for Users: A Psychological Perspective," *KI - Künstliche Intelligenz*, Jul. 17, 2023. DOI: 10.1007/s13218-023-00806-9. [Online]. Available: https://link.springer.com/10.1007/s13218-023-00806-9 (cit. on p. 17).

[38] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance," *Frontiers in Computer Science*, vol. 5, p. 1 096 257, Feb. 6, 2023. DOI: 10.3389/fcomp.2023.1096257. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1096257/full (cit. on pp. 14, 16, 17, 46).

[39] A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek, Eds., *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers* (Lecture Notes in Computer Science). Cham: Springer International Publishing, 2022, vol. 13200. DOI: 10.1007/978-3-031-04083-2. [Online]. Available: https://link.springer.com/10.1007/978-3-031-04083-2 (cit. on pp. 22, 23, 40).

[40] S. Jesus, C. Belém, V. Balayan, *et al.*, "How can I choose an explainer?: An Application-grounded Evaluation of Post-hoc Explanations," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada: ACM, Mar. 3, 2021, pp. 805–815. DOI: 10.1145/3442188.3445941. [Online]. Available: https://dl.acm.org/doi/10.1145/3442188.3445941 (cit. on p. 14).

[41] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, "Foundations for an Empirically Determined Scale of Trust in Automated Systems," *International Journal of Cognitive Ergonomics*, vol. 4, no. 1, pp. 53–71, Mar. 1, 2000. DOI: 10.1207/S15327566IJCE0401_04. [Online]. Available: https://doi.org/10.1207/S15327566IJCE0401_04 (cit. on p. 16).

[42] B. Kim, M. Wattenberg, J. Gilmer, *et al.*, "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)," in *Proceedings of the 35th International Conference on Machine Learning*, 2018. [Online]. Available: http://proceedings.mlr.press/v80/kim18d.html (cit. on pp. 23, 26).

[43] G. Klein, R. R. Hoffman, W. J. Clancey, S. T. Mueller, F. Jentsch, and M. Jalaeian, ""Minimum Necessary Rigor" in empirically evaluating human–AI work systems," *AI Magazine*, vol. 44, no. 3, pp. 274–281, Sep. 2023. DOI: 10.1002/aaai.12108. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/aaai.12108 (cit. on pp. 16, 17, 46).

[44] P. W. Koh, T. Nguyen, Y. S. Tang, *et al.* "Concept Bottleneck Models." arXiv: 2007.04612 [cs, stat]. (Dec. 28, 2020), [Online]. Available: http://arxiv.org/abs/2007.04612, preprint (cit. on pp. 22, 25, 26, 32, 46).

[45] J. Kruschke, "ALCOVE: An Exemplar-Based Connectionist Model of Category Learning," *Psychological review*, vol. 99, pp. 22–44, Jan. 1, 1992. DOI: 10.1037/0033-295X.99.1.22 (cit. on p. 19).

[46] J. Larson, S. Mattu, L. Kirchner, and J. Angwin. "How We Analyzed the COMPAS Recidivism Algorithm," ProPublica. (May 23, 2016), [Online]. Available: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm (cit. on p. 10).

[47] S. Lazar and A. Nelson, "AI safety on whose terms?" *Science*, vol. 381, no. 6654, pp. 138–138, Jul. 14, 2023. DOI: 10.1126/science.adi8982. [Online]. Available: https://www.science.org/doi/10.1126/science.adi8982 (cit. on pp. 11, 15, 16).

[48] S. Lazar, "Power and AI: Nature and Justification," in *The Oxford Handbook of AI Governance*, J. B. Bullock, Y.-C. Chen, J. Himmelreich, *et al.*, Eds., Oxford University Press, 2022. DOI: 10.1093/oxfordhb/9780197579329.013.12. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780197579329.013.12 (cit. on p. 16).

[49] J. D. Lee and K. A. See, "Trust in Automation: Designing for Appropriate Reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, Mar. 1, 2004. DOI: 10.1518/hfes.46.1.50_30392. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1518/hfes.46.1.50_30392 (cit. on p. 16).

[50] B. Leichtmann, C. Humer, A. Hinterreiter, M. Streit, and M. Mara, "Effects of Explainable Artificial Intelligence on trust and human behavior in a high-risk decision task," *Computers in Human Behavior*, vol. 139, p. 107 539, Feb. 2023. DOI: 10.1016/j.chb.2022.107539. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0747563222003594 (cit. on p. 18).

[51] Z. Lipton, "The Mythos of Model Interpretability," *Communications of the ACM*, vol. 61, Oct. 6, 2016. DOI: 10.1145/3233231 (cit. on p. 15).

[52] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," presented at the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA., 2017. [Online]. Available: https://www.researchgate.net/publication/317062430_A_Unified_Approach_to_Interpreting_Model_Predictions (cit. on p. 22).

[53] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and W. Pan. "Promises and Pitfalls of Black-Box Concept Learning Models." arXiv: 2106.13314 [cs]. (Jun. 24, 2021), [Online]. Available: http://arxiv.org/abs/2106.13314, preprint (cit. on pp. 26, 33).

[54] A. Margeloiu, M. Ashman, U. Bhatt, Y. Chen, M. Jamnik, and A. Weller. "Do Concept Bottleneck Models Learn as Intended?" arXiv: 2105.04289 [cs]. (May 10, 2021), [Online]. Available: http://arxiv.org/abs/2105.04289, preprint (cit. on pp. 25, 32).

[55] D. L. Medin, M. M. Schaffer, and B. College, "Context Theory of Classification Learning," *Psychological Review*, vol. 85, no. 3, pp. 207–238, 1978. DOI: 10.1037/0033-295X.85.3.207. [Online]. Available: https://groups.psych.northwestern.edu/medin/documents/MedinSchaffer1978PsychRev.pdf (cit. on p. 19).

[56] T. Miyatsu, R. Gouravajhala, R. M. Nosofsky, and M. A. McDaniel, "Feature highlighting enhances learning of a complex natural-science category.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 45, no. 1, pp. 1–16, Jan. 2019. DOI: 10.1037/xlm0000538. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/xlm0000538 (cit. on pp. 18–20, 31).

[57] S. Mohseni, N. Zarei, and E. D. Ragan, "A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems," *ACM Transactions on Interactive Intelligent Systems*, vol. 11, no. 3-4, pp. 1–45, Dec. 31, 2021. DOI: 10.1145/3387166. [Online]. Available: https://dl.acm.org/doi/10.1145/3387166 (cit. on pp. 14, 15, 17).

[58] R. Zellers, A. Holtzman, H. Rashkin, *et al.*, "Defending against neural fake news," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/3e9f0fc9b2f89e043bc623994Paper.pdf (cit. on p. 13).

[59] R. Nosofsky, "Attention, Similarity, and the Identification-Categorization Relationship," *Journal of experimental psychology. General*, vol. 115, pp. 39–61, Mar. 1, 1986. DOI: 10.1037/0096-3445.115.1.39 (cit. on p. 19).

[60] R. M. Nosofsky, C. A. Sanders, B. J. Meagher, and B. J. Douglas, "Toward the development of a feature-space representation for a complex natural category domain," *Behavior Research Methods*, vol. 50, no. 2, pp. 530–556, Apr. 2018. DOI: 10.3758/s13428-017-0884-8. [Online]. Available: http://link.springer.com/10.3758/s13428-017-0884-8 (cit. on pp. 7, 31).

[61] R. M. Nosofsky, C. A. Sanders, A. Gerdom, B. J. Douglas, and M. A. McDaniel, "On Learning Natural-Science Categories That Violate the Family-Resemblance Principle," *Psychological Science*, vol. 28, no. 1, pp. 104–114, Jan. 2017. DOI: 10.1177/0956797616675636. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0956797616675636 (cit. on p. 40).

[62] R. M. Nosofsky, C. A. Sanders, and M. A. McDaniel, "Tests of an exemplar-memory model of classification learning in a high-dimensional natural-science category domain.," *Journal of Experimental Psychology: General*, vol. 147, no. 3, pp. 328–353, Mar. 2018. DOI: 10.1037/xge0000369. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/xge0000369 (cit. on p. 18).

[63] OpenAI. "Jukebox." (2023), [Online]. Available: https://openai.com/research/jukebox (cit. on p. 11).

[64] E. Pashentsev, Ed., *The Palgrave Handbook of Malicious Use of AI and Psychological Security*. Cham: Springer International Publishing, 2023. DOI: 10.1007/978-3-031-22552-9. [Online]. Available: https://link.springer.com/10.1007/978-3-031-22552-9 (cit. on p. 11).

[65] S. A. C. Perrig, N. Scharowski, and F. Brühlmann, "Trust Issues with Trust Scales: Examining the Psychometric Quality of Trust Measures in the Context of AI," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg Germany: ACM, Apr. 19, 2023, pp. 1–7. DOI: 10.1145/3544549.3585808. [Online]. Available: https://dl.acm.org/doi/10.1145/3544549.3585808 (cit. on pp. 16, 17, 46).

[66] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," *Proceedings of the 38 th International Conference on Machine Learning, PMLR 139, 2021*, 2021. DOI: 10.48550/arXiv.2103.00020 (cit. on p. 27).

[67] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, "Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, Raleigh, NC, USA: IEEE, Feb. 2023, pp. 464–483. DOI: 10.1109/SaTML54575.2023.00039. [Online]. Available: https://ieeexplore.ieee.org/document/10136140/ (cit. on p. 41).

[68] M. T. Ribeiro, S. Singh, and C. Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." arXiv: 1602.04938 [cs, stat]. (Aug. 9, 2016), [Online]. Available: http://arxiv.org/abs/1602.04938, preprint (cit. on pp. 18, 22).

[69] C. Rudin, C. Wang, and B. Coker, "The Age of Secrecy and Unfairness in Recidivism Prediction," *Harvard Data Science Review*, vol. 2, no. 1, Jan. 31, 2020. DOI: 10.1162/99608f92.6ed64b30. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/7z10o269 (cit. on p. 11).

[70] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. "Interpretable Machine Learning: Fundamental Principles and 10 Grand Challenges." arXiv: 2103.11251 [cs, stat]. (Jul. 9, 2021), [Online]. Available: http://arxiv.org/abs/2103.11251, preprint (cit. on p. 22).

[71] C. Rudin. "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead." arXiv: 1811.10154 [cs, stat]. (Sep. 21, 2019), [Online]. Available: http://arxiv.org/abs/1811.10154, preprint (cit. on pp. 23, 26).

[72] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (Lecture Notes in Computer Science). Cham: Springer International Publishing, 2019, vol. 11700. DOI: 10.1007/978-3-030-28954-6. [Online]. Available: http://link.springer.com/10.1007/978-3-030-28954-6 (cit. on p. 14).

[73] C. A. Sanders and R. M. Nosofsky, "Training Deep Networks to Construct a Psychological Feature Space for a Natural-Object Category Domain," *Computational Brain & Behavior*, vol. 3,

no. 3, pp. 229–251, Sep. 2020. DOI: 10.1007/s42113-020-00073-z. [Online]. Available: http://link.springer.com/10.1007/s42113-020-00073-z (cit. on pp. 18–20, 31).

[74] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74. [Online]. Available: http://ieeexplore.ieee.org/document/8237336/ (cit. on pp. 18, 22).

[75] L. S. Shapley, "17. A Value for n-Person Games," in *Contributions to the Theory of Games (AM-28), Volume II*, H. W. Kuhn and A. W. Tucker, Eds., Princeton University Press, Dec. 31, 1953, pp. 307–318. DOI: 10.1515/9781400881970-018. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/9781400881970-018/html (cit. on p. 28).

[76] K. Simonyan, A. Vedaldi, and A. Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." arXiv: 1312.6034 [cs]. (Apr. 19, 2014), [Online]. Available: http://arxiv.org/abs/1312.6034, preprint (cit. on p. 22).

[77] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. "SmoothGrad: Removing noise by adding noise." arXiv: 1706.03825 [cs, stat]. (Jun. 12, 2017), [Online]. Available: http://arxiv.org/abs/1706.03825, preprint (cit. on pp. 22, 23).

[78] D. J. Smith and J. P. Minda, "Thirty categorization results in search of a model.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 26, no. 1, pp. 3–27, 2000. DOI: 10.1037/0278-7393.26.1.3. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/0278-7393.26.1.3 (cit. on p. 19).

[79] *SoftVC VITS Singing Voice Conversion*, MoeVoiceConversion, Aug. 29, 2023. [Online]. Available: https://github.com/svc-develop-team/so-vits-svc (cit. on p. 11).

[80] "South Korea's Presidential Deepfake." (Mar. 29, 2022), [Online]. Available: https://www.vastmindz.com/south-koreas-presidential-deepfake/ (cit. on p. 11).

[81] F. Sovrano and F. Vitali, "Explanatory artificial intelligence (YAI): Human-centered explanations of explainable AI and complex data," *Data Mining and Knowledge Discovery*, Oct. 10, 2022. DOI: 10.1007/s10618-022-00872-x. [Online]. Available: https://link.springer.com/10.1007/s10618-022-00872-x (cit. on p. 14).

[82] "Speechify | Complete AI Voice Studio | TTS, AI Voice Over & More." (May 4, 2022), [Online]. Available: https://speechify.com/ (cit. on p. 11).

[83] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An Open Multilingual Graph of General Knowledge," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 12, 2017. DOI: 10.1609/aaai.v31i1.11164. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11164 (cit. on p. 28).

[84] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning, PMLR 70:3319-3328, 2017*, 2017. [Online]. Available: http://proceedings.mlr.press/v70/sundararajan17a.html (cit. on p. 22).

[85] T. Ueno, Y. Sawa, Y. Kim, J. Urakami, H. Oura, and K. Seaborn, "Trust in Human-AI Interaction: Scoping Out Models, Measures, and Methods," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, New Orleans LA USA: ACM, Apr. 27, 2022, pp. 1–7. DOI: 10.1145/3491101.3519772. [Online]. Available: https://dl.acm.org/doi/10.1145/3491101.3519772 (cit. on pp. 16, 17, 46).

[86] J.-W. Van Prooijen and M. Acker, "The Influence of Control on Belief in Conspiracy Theories: Conceptual and Applied Extensions: Control and conspiracy belief," *Applied Cognitive Psychology*, vol. 29, no. 5, pp. 753–761, Sep. 2015. DOI: 10.1002/acp.3161. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/acp.3161 (cit. on pp. 12, 13).

[87] J.-W. Van Prooijen, "Why Education Predicts Decreased Belief in Conspiracy Theories: Education and Conspiracy Beliefs," *Applied Cognitive Psychology*, vol. 31, no. 1, pp. 50–58, Jan. 2017. DOI: 10.1002/acp.3301. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/acp.3301 (cit. on pp. 12, 13).

[88] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," 2011. [Online]. Available: https://authors.library.caltech.edu/27452/1/CUB_200_2011.pdf (cit. on p. 20).

[89] W. H. Walters and E. I. Wilder, "Fabrication and errors in the bibliographic citations generated by ChatGPT," *Scientific Reports*, vol. 13, no. 1, p. 14 045, Sep. 7, 2023. DOI: 10.1038/s41598-

023-41032-5. [Online]. Available: https://www.nature.com/articles/s41598-023-41032-5 (cit. on p. 12).

[90] F. M. Walter, A. T. Prevost, J. Vasconcelos, *et al.*, "Using the 7-point checklist as a diagnostic aid for pigmented skin lesions in general practice: A diagnostic validation study," *British Journal of General Practice*, vol. 63, no. 610, e345–e353, May 2013. DOI: 10.3399/bjgp13X667213. [Online]. Available: https://bjgp.org/lookup/doi/10.3399/bjgp13X667213 (cit. on p. 25).

[91] S. Wang, Z. Zhao, X. Ouyang, Q. Wang, and D. Shen. "ChatCAD: Interactive Computer-Aided Diagnosis on Medical Image using Large Language Models." arXiv: 2302.07257 [cs, eess]. (Feb. 14, 2023), [Online]. Available: http://arxiv.org/abs/2302.07257, preprint (cit. on pp. 14, 18, 29, 30).

[92] K. Wiggers. "New York City announces task force to find biases in algorithms," VentureBeat. (May 16, 2018), [Online]. Available: https://venturebeat.com/ai/new-york-city-announces-task-force-to-find-biases-in-algorithms/ (cit. on p. 12).

[93] W. E. Forum, *The Presidio Recommendations on Responsible Generative AI*, Jun. 2023. [Online]. Available: https://www3.weforum.org/docs/WEF_Presidio_Recommendations_on_Responsible_Generative_AI_2023.pdf (cit. on pp. 7, 13).

[94] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua. "NExT-GPT: Any-to-Any Multimodal LLM." arXiv: 2309.05519 [cs]. (Sep. 13, 2023), [Online]. Available: http://arxiv.org/abs/2309.05519, preprint (cit. on p. 30).

[95] Y. Yang, A. Panagopoulou, S. Zhou, D. Jin, C. Callison-Burch, and M. Yatskar, "Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023*, 2023. DOI: 10.48550/arXiv.2211.11158 (cit. on pp. 29, 30).

[96] W. Yang, Y. Wei, H. Wei, *et al.*, "Survey on Explainable AI: From Approaches, Limitations and Applications Aspects," *Human-Centric Intelligent Systems*, vol. 3, no. 3, pp. 161–188, Aug. 10, 2023. DOI: 10.1007/s44230-023-00038-y. [Online]. Available: https://link.springer.com/10.1007/s44230-023-00038-y (cit. on p. 22).

[97] C.-K. Yeh, B. Kim, S. Ö. Arık, C.-L. Li, T. Pfister, and P. Ravikumar, "On Completeness-aware Concept-Based Explanations in Deep Neural Networks," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2019. DOI: 10.48550/arXiv.1910.07969. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/ecb287ff763c169694f682af52c1f309-Paper.pdf (cit. on pp. 28, 43, 45).

[98] M. Yuksekgonul, M. Wang, and J. Zou. "Post-hoc Concept Bottleneck Models." arXiv: 2205.15480 [cs, stat]. (2022), [Online]. Available: http://arxiv.org/abs/2205.15480, preprint (cit. on pp. 25–27, 33).

[99] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8689, Cham: Springer International Publishing, 2014, pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10590-1_53 (cit. on p. 22).

[100] B. Zhang and A. Dafoe, "Artificial Intelligence: American Attitudes and Trends," *SSRN Electronic Journal*, 2019. DOI: 10.2139/ssrn.3312874. [Online]. Available: https://www.ssrn.com/abstract=3312874 (cit. on p. 14).

[101] T. Zhang, X. J. Yang, and B. Li. "May I Ask a Follow-up Question? Understanding the Benefits of Conversations in Neural Network Explainability." arXiv: 2309.13965 [cs]. (Sep. 25, 2023), [Online]. Available: http://arxiv.org/abs/2309.13965, preprint (cit. on pp. 14, 17, 29).

[102] H. Zhao, H. Chen, F. Yang, *et al.* "Explainability for Large Language Models: A Survey." arXiv: 2309.01029 [cs]. (Sep. 16, 2023), [Online]. Available: http://arxiv.org/abs/2309.01029, preprint (cit. on pp. 17, 29).

[103] B. Zhou, D. Bau, A. Oliva, and A. Torralba, "Interpreting Deep Visual Representations via Network Dissection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2131–2145, 2017. DOI: 10.1109/TPAMI.2018.2858759. [Online]. Available: https://ieeexplore.ieee.org/document/8417924/ (cit. on p. 24).

[104] A. Zimmermann, K. Vredenburgh, and S. Lazar, "The Political Philosophy of Data and AI," *Canadian Journal of Philosophy*, vol. 52, no. 1, pp. 1–5, Jan. 2022. DOI: 10.1017/can.

2022.28. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0045509122000285/type/journal_article (cit. on p. 16).

[105] M. Schrum, M. Ghuy, E. Hedlund-botti, M. Natarajan, M. Johnson, and M. Gombolay, "Concerning Trends in Likert Scale Usage in Human-robot Interaction: Towards Improving Best Practices," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–32, Sep. 30, 2023. DOI: 10.1145/3572784. [Online]. Available: https://dl.acm.org/doi/10.1145/3572784 (cit. on p. 16).

[106] T. M. Inc., *Matlab version: 9.14.0 (r2023a)*, Natick, Massachusetts, United States, 2023. [Online]. Available: https://www.mathworks.com (cit. on p. 32).

[107] P. T. Inc. "Collaborative data science." (2015), [Online]. Available: https://plot.ly (cit. on p. 36).

[108] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, Dec. 2014. DOI: 10.48550/arXiv.1412.6980. eprint: 1412.6980 (cit. on p. 37).

Further Reading

[1] G. Alain and Y. Bengio. "Understanding intermediate layers using linear classifier probes." arXiv: 1610.01644 [cs, stat]. (Nov. 22, 2018), [Online]. Available: http://arxiv.org/abs/1610.01644, preprint.

[2] V. Balayan, P. Saleiro, C. Belém, L. Krippahl, and P. Bizarro. "Teaching the Machine to Explain Itself using Domain Knowledge." arXiv: 2012.01932 [cs]. (Nov. 27, 2020), [Online]. Available: http://arxiv.org/abs/2012.01932, preprint.

[3] A. Begum, S. Alex David, D. Hemalatha, and L. S. S. Kollipara, "Deep Learning-Based Lung Cancer Classification: Recent Developments and Future Prospects," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India: IEEE, May 25, 2023, pp. 1–8. DOI: 10.1109/ACCAI58221.2023.10200967. [Online]. Available: https://ieeexplore.ieee.org/document/10200967/.

[4] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Computers & Chemical Engineering*, vol. 108, pp. 250–267, Jan. 2018. DOI: 10.1016/j.compchemeng.2017.09.017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0098135417303228.

[5] J. Cheng and M. S. Bernstein, "Flock: Hybrid Crowd-Machine Learning Classifiers," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, Vancouver BC Canada: ACM, Feb. 28, 2015, pp. 600–611. DOI: 10.1145/2675133.2675214. [Online]. Available: https://dl.acm.org/doi/10.1145/2675133.2675214.

[6] D. Doran, S. Schulz, and T. R. Besold. "What Does Explainable AI Really Mean? A New Conceptualization of Perspectives." arXiv: 1710.00794 [cs]. (Oct. 2, 2017), [Online]. Available: http://arxiv.org/abs/1710.00794, preprint.

[7] F. Doshi-Velez and B. Kim. "Towards A Rigorous Science of Interpretable Machine Learning." arXiv: 1702.08608 [cs, stat]. (Mar. 2, 2017), [Online]. Available: http://arxiv.org/abs/1702.08608, preprint.

[8] R. W. Fleming and K. R. Storrs, "Learning to see stuff," *Current Opinion in Behavioral Sciences*, vol. 30, pp. 100–108, Dec. 2019. DOI: 10.1016/j.cobeha.2019.07.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352154619300397.

[9] R. W. Fleming, "Material Perception," *Annu Rev Vision Sci*, 2017. DOI: 10.1146/annurev-vision-102016-061429.

[10] A. Ghorbani and J. Zou, "Neuron Shapley: Discovering the Responsible Neurons," *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/41c542dfe6e4fc3deb251d64cf6ed2e4-Abstract.html.

[11] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design," *Journal of Machine Learning Research*, vol. 11, 2010. [Online]. Available: https://www.jmlr.org/papers/volume11/gorissen10a/gorissen10a.pdf.

[12] C. G. Gross, "Genealogy of the "Grandmother Cell"," *The Neuroscientist*, vol. 8, no. 5, pp. 512–518, Oct. 2002. DOI: 10.1177/107385802237175. [Online]. Available: http://journals.sagepub.com/doi/10.1177/107385802237175.

[13] D. Gunning and D. W. Aha, "DARPA's Explainable Artificial Intelligence Program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, Jun. 2019. DOI: 10.1609/aimag.v40i2.2850. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1609/aimag.v40i2.2850.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/.

[15] M. Hind, D. Wei, M. Campbell, *et al.*, "TED: Teaching AI to Explain its Decisions," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, Honolulu HI USA: ACM, Jan. 27, 2019, pp. 123–129. DOI: 10.1145/3306618.3314273. [Online]. Available: https://dl.acm.org/doi/10.1145/3306618.3314273.

[16] R. R. Hoffman and G. Klein, "Explaining Explanation, Part 1: Theoretical Foundations," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 68–73, May 2017. DOI: 10.1109/MIS.2017.54. [Online]. Available: http://ieeexplore.ieee.org/document/7933919/.

[17] R. R. Hoffman, S. T. Mueller, and G. Klein, "Explaining Explanation, Part 2: Empirical Foundations," *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 78–86, 2017. DOI: 10.1109/MIS.2017.3121544. [Online]. Available: http://ieeexplore.ieee.org/document/8012316/.

[18] R. Hoffman, T. Miller, S. T. Mueller, G. Klein, and W. J. Clancey, "Explaining Explanation, Part 4: A Deep Dive on Deep Nets," *IEEE Intelligent Systems*, vol. 33, no. 3, pp. 87–95, May 2018. DOI: 10.1109/MIS.2018.033001421. [Online]. Available: https://ieeexplore.ieee.org/document/8423529/.

[19] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A Benchmark for Interpretability Methods in Deep Neural Networks," 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/hash/fe4b8556000d0f0cae99daa5c5c5a410-Abstract.html.

[20] H. Khosravi, S. B. Shum, G. Chen, *et al.*, "Explainable Artificial Intelligence in education," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100 074, 2022. DOI: 10.1016/j.caeai.2022.100074. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2666920X22000297.

[21] G. Klein, "Explaining Explanation, Part 3: The Causal Landscape," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 83–88, Mar. 2018. DOI: 10.1109/MIS.2018.022441353. [Online]. Available: https://ieeexplore.ieee.org/document/8378482/.

[22] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th International Conference on Computer Vision*, Kyoto: IEEE, Sep. 2009, pp. 365–372. DOI: 10.1109/ICCV.2009.5459250. [Online]. Available: http://ieeexplore.ieee.org/document/5459250/.

[23] I. Lage and F. Doshi-Velez. "Learning Interpretable Concept-Based Models with Human Feedback." arXiv: 2012.02898 [cs, stat]. (Dec. 4, 2020), [Online]. Available: http://arxiv.org/abs/2012.02898, preprint.

[24] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, Jun. 2009, pp. 951–958. DOI: 10.1109/CVPR.2009.5206594. [Online]. Available: https://ieeexplore.ieee.org/document/5206594/.

[25] C. Liao, M. Sawayama, and B. Xiao, "Unsupervised learning reveals interpretable latent representations for translucency perception," *PLOS Computational Biology*, vol. 19, no. 2, R. W. Fleming, Ed., e1010878, Feb. 8, 2023. DOI: 10.1371/journal.pcbi.1010878. [Online]. Available: https://dx.plos.org/10.1371/journal.pcbi.1010878.

[26] M. Losch, M. Fritz, and B. Schiele. "Interpretability Beyond Classification Output: Semantic Bottleneck Networks." arXiv: 1907.10882 [cs]. (Jul. 28, 2019), [Online]. Available: http://arxiv.org/abs/1907.10882, preprint.

[27] A. L. Martel, P. Abolmaesumi, D. Stoyanov, *et al.*, Eds., *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I* (Lecture Notes in Computer Science). Cham: Springer International Publishing, 2020, vol. 12261. DOI: 10.1007/978-3-030-59710-8. [Online]. Available: https://link.springer.com/10.1007/978-3-030-59710-8.

[28] D. Martens, C. Dams, J. Hinns, and M. Vergouwen. "Tell Me a Story! Narrative-Driven XAI with Large Language Models." arXiv: 2309.17057 [cs]. (Sep. 29, 2023), [Online]. Available: http://arxiv.org/abs/2309.17057, preprint.

[29] N. Martin, "Selectivity in Neural Networks," University of Bristol, 2021. [Online]. Available: https://research-information.bris.ac.uk/en/studentTheses/selectivity-in-neural-networks.

[30] M. Mehta, V. Palade, and I. Chatterjee, Eds., *Explainable AI: Foundations, Methodologies and Applications* (Intelligent Systems Reference Library). Cham: Springer International Publishing, 2023, vol. 232. DOI: 10.1007/978-3-031-12807-3. [Online]. Available: https://link.springer.com/10.1007/978-3-031-12807-3.

[31] D. A. Melis and T. Jaakkola, "Towards Robust Interpretability with Self-Explaining Neural Networks," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.*, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html.

[32] Meta. "Enforcing Against Manipulated Media," Meta. (Jan. 7, 2020), [Online]. Available: https://about.fb.com/news/2020/01/enforcing-against-manipulated-media/.

[33] T. N. Mundhenk, B. Y. Chen, and G. Friedland. "Efficient Saliency Maps for Explainable AI." arXiv: 1911.11293 [cs]. (Mar. 9, 2020), [Online]. Available: http://arxiv.org/abs/1911.11293, preprint.

[34] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*, 2. ed., reprinted. Amsterdam: Acad. Press, 2010, 406 pp. [Online]. Available: https://ds.amu.edu.et/xmlui/bitstream/handle/123456789/14547/feature-extraction-image-processing-second-ed%20-%20423%20pages.pdf?sequence=1&isAllowed=y.

[35] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-Precision Model-Agnostic Explanations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 25, 2018. DOI: 10.1609/aaai.v32i1.11491. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11491.

[36] M. T. Ribeiro, S. Singh, and C. Guestrin. "Model-Agnostic Interpretability of Machine Learning." arXiv: 1606.05386 [cs, stat]. (Jun. 16, 2016), [Online]. Available: http://arxiv.org/abs/1606.05386, preprint.

[37] D. A. Scheufele and N. M. Krause, "Science audiences, misinformation, and fake news," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7662–7669, Apr. 16, 2019. DOI: 10.1073/pnas.1805871115. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1805871115.

[38] Y. Shen, L. Wang, Y. Chen, X. Xiao, J. Liu, and H. Wu. "An Interpretability Evaluation Benchmark for Pre-trained Language Models." arXiv: 2207.13948 [cs]. (Jul. 28, 2022), [Online]. Available: http://arxiv.org/abs/2207.13948, preprint.

[39] S. Shin, Y. Jo, S. Ahn, and N. Lee. "A Closer Look at the Intervention Procedure of Concept Bottleneck Models." arXiv: 2302.14260 [cs]. (Jul. 2, 2023), [Online]. Available: http://arxiv.org/abs/2302.14260, preprint.

[40] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. "Striving for Simplicity: The All Convolutional Net." arXiv: 1412.6806 [cs]. (Apr. 13, 2015), [Online]. Available: http://arxiv.org/abs/1412.6806, preprint.

[41] A. Weller. "Transparency: Motivations and Challenges." arXiv: 1708.01870 [cs]. (Aug. 19, 2019), [Online]. Available: http://arxiv.org/abs/1708.01870, preprint.

[42] B. Zhang, "Public Opinion toward Artificial Intelligence," 2022. [Online]. Available: https://osf.io/download/615efb1642b47400dc00e916/.

[43] W. X. Zhao, K. Zhou, J. Li, *et al.* "A Survey of Large Language Models." arXiv: 2303.18223 [cs]. (Jun. 29, 2023), [Online]. Available: http://arxiv.org/abs/2303.18223, preprint.

[44] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable Basis Decomposition for Visual Explanation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11212, Cham: Springer International Publishing, 2018, pp. 122–138. DOI: 10.1007/978-3-030-01237-3_8. [Online]. Available: https://link.springer.com/10.1007/978-3-030-01237-3_8.

[45] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, Jun. 1, 2018. DOI: 10.1109/TPAMI.2017.2723009. [Online]. Available: https://ieeexplore.ieee.org/document/7968387/.

[46] B. Zhou, Y. Sun, D. Bau, and A. Torralba. "Revisiting the Importance of Individual Units in CNNs via Ablation." arXiv: 1806.02891 [cs]. (Jun. 7, 2018), [Online]. Available: http://arxiv.org/abs/1806.02891, preprint.